

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

---

**ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ  
УПРАВЛЕНИЯ (ИТТСУ)**

**Кафедра «Управление и защита информации»**

**А.И. САФРОНОВ**

**ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ ВИРТУАЛЬНЫХ  
ПРИБОРОВ *NATIONAL INSTRUMENTS LABVIEW***

**Сборник типовых задач  
для проведения аудиторных занятий по Учебной практике**

**МОСКВА – 2021**

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

---

ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ  
УПРАВЛЕНИЯ (ИТТСУ)

Кафедра «Управление и защита информации»

А.И. САФРОНОВ  
ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ ВИРТУАЛЬНЫХ  
ПРИБОРОВ *NATIONAL INSTRUMENTS LABVIEW*

Сборник типовых задач  
для магистрантов направления  
27.04.04 «Управление в технических системах»

**МОСКВА – 2021**

**УДК 004**

**С 21**

Сафронов А.И. Проектирование и создание виртуальных приборов *National Instruments LabView*: Сборник типовых задач для проведения аудиторных занятий по Учебной практике. – М.: РУТ (МИИТ), 2021. – 181 с.

Сборник типовых задач содержит постановки к работам, которые обучающиеся должны выполнить для приобретения навыков использования пакета прикладных программ и среды разработки виртуальных приборов *National Instruments LabView*. *NI LabView* является передовой разработкой в области информационных технологий, позволяющей выполнять моделирование как чисто программных, так и программно-аппаратных звеньев автоматизированных систем управления на базе типовых микропроцессорных устройств, а также специализированных микропроцессорных модулей, предназначенных для сложных оптимизационных расчётов, проводимых в реальном времени. Сборник задач ориентирован на рассмотрение программных реализаций, предоставляя обучающимся возможность понять внутреннее устройство решений, внедрённых в учебный процесс на кафедре «Управление и защита информации». Приобретённые навыки способствуют закреплению основ программирования и алгоритмизации, развитию образного мышления, эффективному восприятию принципиальных технических схем, пониманию потоковых вычислений и элементов параллельного программирования.

© РУТ (МИИТ), 2021

## Введение

Данный Сборник типовых задач содержит постановки к трём объёмным инженерным, практическим заданиям, необходимым для развития у обучающихся первичных профессиональных умений и навыков, в том числе, первичных умений и навыков научно-исследовательской деятельности.

К числу первичных умений и навыков научно-исследовательской деятельности стоит отнести отработку на «Учебной практике»:

- аналитического вывода формул,
- построения алгоритмов решения инженерных задач,
- поиска способов решения инженерных задач,
- подбора соответствующих методов, обоснования принятых решений,
- формулировки развёрнутого и содержательного вывода о проделанной работе.

Представленные в Сборнике задания рассчитаны на большую группу студентов, обучающихся по направлению 27.04.04 «Управление в технических системах» (магистратура), численностью до 35 человек.

Сборник типовых задач является учебно-методическим дополнением действующей и утверждённой программы практик, проводимых на кафедре «Управление и защита информации», а также за её пределами, на предприятиях [1].

Счёт заданий, изложенных в настоящем Сборнике, начинается с девятки («9»). Подразумевается, что в качестве первого задания «Учебной практики» обучающиеся выполняют отчётную документацию к решённой задаче алгоритмизации и программирования, постановка которой изложена учебно-методическом пособии [2], а задания со второго по восьмое содержатся в Сборнике задач [3] и позволяют обучающимся закрепить навыки работы в электронных таблицах *Microsoft Office Excel*.

В связи с отмеченными выше положениями обучающимся необходимо внимательно следить за нумерацией, указанной в постановочных частях заданий Сборника типовых задач, поскольку реальные номера этих заданий, согласно программе «Учебной практики» [1], отличаются от номеров параграфов (разделов), в которых они изложены в этом издании.

Задания, входящие в состав Сборника типовых задач, систематизированы по следующей опорной схеме:

- наименование задания (указано в заголовке раздела первого уровня);
- постановочный раздел задания, содержащий формулировку общей части работы для группы обучающихся (общая часть задания выполняется всеми обучающимися без исключения до выполнения индивидуальных задач и должна быть оформлена в соответствующем разделе отчёта; для некоторых заданий Сборника общая часть может отсутствовать);
- формулировка цели работы;
- полезные соотношения, необходимые для выполнения работы (актуально не для всех заданий Сборника);
- пример выполнения задания или его фрагмента (актуально не для всех заданий Сборника);
- варианты индивидуального задания;
- образец компоновки графического пользовательского интерфейса виртуального прибора (далее – ВП);
- образец базовой разметки исходных данных на блок-диаграмме ВП;
- схема алгоритма функционирования типового ВП, создаваемого в рамках работы над текущим рассматриваемым заданием.

Сборник типовых задач разработан с целью повышения познавательной активности обучающихся, достигаемой за счёт уместных отсылок к справочной литературе, а также ранее пройденному материалу в курсах:

- «Высшей математики» [4],
- «Программирования и основ алгоритмизации»,
- «Информатики»,
- «Физики» [5][6],
- «Теоретических основ электротехники» [4][7][8][9][10],
- «Электроники» [10],
- «Микропроцессорных устройств систем управления» [6],
- «Численных методов в инженерных задачах» [8],
- «Компьютерной математики» [4],
- «Цифровой обработки сигналов» [5][8][11],
- «Технических средств автоматизации и управления» [6][10],
- и других.

Задания, с точки зрения алгоритмизации, тематически ограничены рассмотрением только последовательного и разветвляющегося вычислительных процессов, а с точки зрения программирования рассмотрением:

- численных элементов управления («*Numeric*»);
- логических элементов управления («*Boolean*»);
- элементов сравнения («*Comparison*»);
- структур («*Structures*») разветвления вычислительного процесса.

Отчёты, оформляемые по результатам выполнения заданий программирования и алгоритмизации, должны соответствовать шаблону, представленному в Приложении 1. В Приложении 2 сформулированы рекомендации, позволяющие обучающимся повысить качество подготавливаемых отчётов. В Приложении 3 собран перечень требований, предъявляемых к именам файлов, направляемых на проверку по электронной почте (элементы дистанционного обучения [1]).

# 1 Начало работы с пакетом прикладных программ *National Instruments LabView*

## 1.1 Переход к созданию нового ВП

Для выполнения всех поставленных заданий данного Сборника предполагается работа обучающихся в пакете прикладных программ *National Instruments LabView* версии 6.1<sup>1</sup>. В разделах, содержащих полезные соотношения для выполнения заданий, рассматриваются составляющие (интерфейс, основные разделы меню, элементы управления и другие) именно этой версии программного обеспечения.

В качестве альтернативы для обучающихся, привыкших решать задачи не по шаблонам, предлагается работа в пакете прикладных программ *National Instruments LabView* выпуска 2016 года<sup>2</sup>.

Представленные по сноске гиперссылки содержат апробированное программное обеспечение, укомплектованное всеми необходимыми компонентами для его установки на персональные компьютеры, работающие под управлением современных операционных систем семейства *Microsoft Windows*, и последующей нормальной работы на протяжении длительного времени без каких-либо ограничений.

Процесс установки программного обеспечения на персональный компьютер в Сборнике не рассматривается. Дальнейшее изложение материала предполагает, что пакет прикладных программ *NI LabView* версии 6.1 уже предустановлен на персональном компьютере, но ещё ни разу не запускался.

После запуска исполняемого файла пакета прикладных программ *National Instruments LabView 6.1* открывается (если настройками не предусмотрено иного) диалоговое окно (Рисунок 1.1), в котором можно:

---

<sup>1</sup> <https://yadi.sk/d/JRhS6ITDqHZqC>

<sup>2</sup> <https://drive.google.com/open?id=18EHK-h-3kCJtAuDELiijaligTm5r8ohQ>

- приступить к созданию нового ВП («*New VI*»);
- продолжить разработку ранее созданного ВП («*Open VI*»);
- перейти к готовым решениям по сбору данных («*DAQ (Data Acquisition) Solutions*»); при первом обращении запускает «мастер настройки устройств сбора информации» – *DAQ hardware Wizard*);
- перейти к справочной подсистеме поиска примеров реализации ВП («*Find Examples*»), входящих в состав пакета прикладных программ *NI LabView*;
- приступить к изучению справочных материалов («*LabView Tutorial*»), позволяющих быстро начать работу с пакетом прикладных программ *National Instruments LabView 6.1* при использовании другой справочной подсистемы;
- завершить работу с диалоговым окном («*Exit*»).

Представленное описание относится к набору кнопок, расположенных в правой части диалогового окна. Левая часть диалогового окна (Рисунок 1.1.1) содержит логотип *NI LabView 6.1*, быстрые подсказки по работе с пакетом прикладных программ, а также область настройки формата вывода диалогового окна начала работы с *NI LabView*.

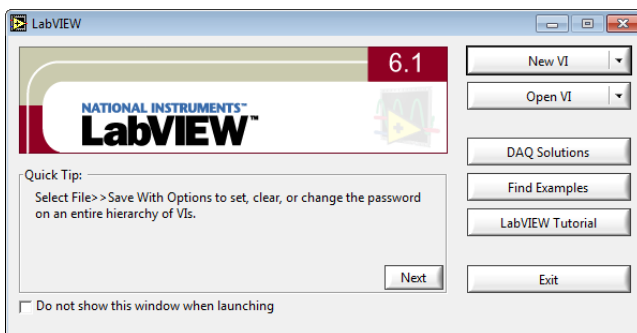


Рисунок 1.1.1 – Начальное диалоговое окно пакета прикладных программ *National Instruments LabView 6.1*. Первый запуск программного обеспечения



Диалоговое окно начала работы с пакетом прикладных программ, запущенное впервые на персональном компьютере, отличается возможностью выставления в нём логического признака, запрещающего вывод аналогичного диалогового окна при последующих запусках программного обеспечения («*Do not show this window when launching*»).

Если логический признак не был выставлен при первом запуске программного обеспечения, аналогичное диалоговое окно будет демонстрироваться каждый раз при обращении к исполняемому файлу *NI LabView*. Вместо логического признака на нём будет размещаться тумблер, переключающий развёрнутый диалог («*Large Dialog*») на краткий диалог («*Small Dialog*») и обратно (Рисунок 1.1.2).

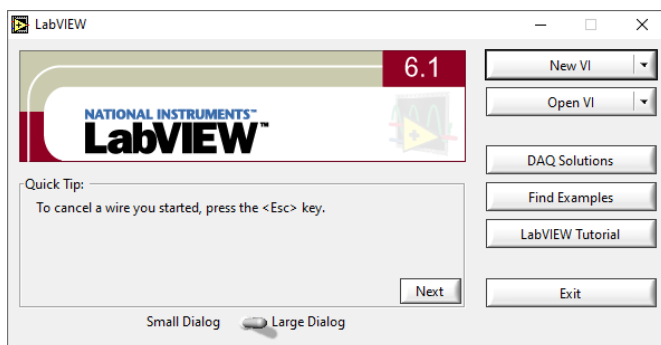


Рисунок 1.1.2 – Начальное диалоговое окно пакета прикладных программ *National Instruments LabView 6.1*. Последующие запуски программного обеспечения. Развёрнутый диалог

Краткий диалог (Рисунок 1.1.3) отличается от развёрнутого тем, что из него исключены все элементы справочных подсистем, а также элементы связи со внешними системами сбора данных (датчиками).

Материалы, представленные далее в Сборнике типовых задач, всецело ориентированы на использование краткого диалога. Рассматривается разработка программного, а не программно-аппаратного обеспечения, а также в качестве

справочного материала в Сборнике выполнено описание элементов управления, необходимых для разработки учебных ВП. Полагается, что обращение обучающихся к внутренним справочным подсистемам *NI LabView 6.1* при решении задач, представленных в Сборнике, избыточно.

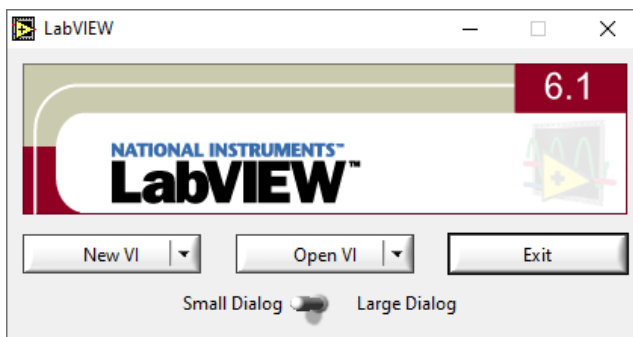


Рисунок 1.1.3 – Начальное диалоговое окно пакета прикладных программ *National Instruments LabView 6.1*. Последующие запуски программного обеспечения. Краткий диалог

Для начала работы по созданию нового ВП необходимо нажать на кнопку «*New VI*». Сокращение «*VI*» в данном случае является обозначением самого ВП (то есть «*Virtual Instrument*» в англоязычной литературе, «Виртуальный Прибор» – в русскоязычной литературе).

Выбор данного режима автоматически генерирует два рабочих окна, каждое из которых предназначено для реализации конкретных составляющих ВП и имеет собственное наименование. Эти окна подробно рассматриваются далее в разделе 1.2.

## 1.2 Описание рабочих окон ВП: передней панели и блок-диаграммы

Как правило, первым отображается окно, именуемое передней / лицевой панелью («*Front Panel*») ВП. В нём проектируется графический пользовательский интерфейс ВП, под ним размещается другое окно, именуемое блок-диаграммой кода («*Block Diagram*») ВП или просто блок-диаграммой ВП. Окно передней панели ВП представлено на Рисунке 1.2.1.

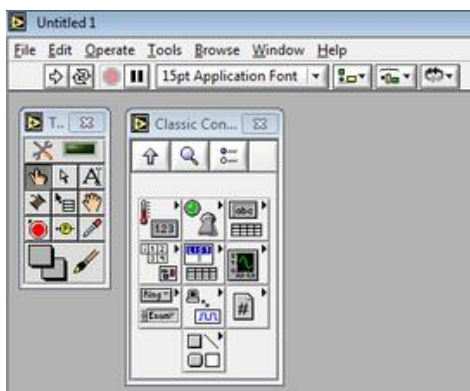


Рисунок 1.2.1 – Окно проектирования лицевой панели ВП в пакете прикладных программ *National Instruments LabView 6.1*

Для быстрого перехода от передней панели к блок-диаграмме и обратно опытные пользователи применяют сочетание клавиш «*Ctrl*» + «*E*», менее опытные – используют переход через пункты меню «*Window* > *Show Diagram*», как показано на Рисунке 1.2.2.

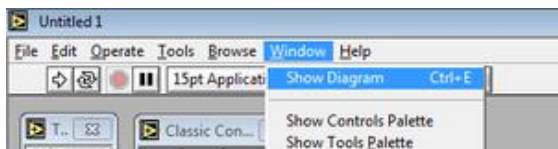


Рисунок 1.2.2 – Переход к блок-диаграмме ВП через пункт меню «*Window*»

На Рисунке 1.2.3 показано окно блок-диаграммы нового ВП.

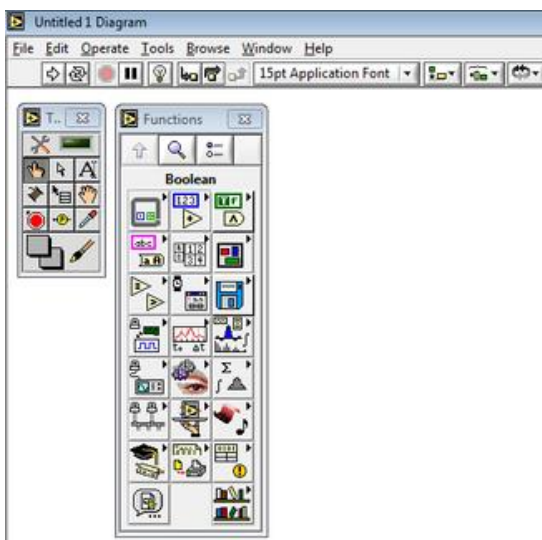


Рисунок 1.2.3 – Окно блок-диаграммы нового ВП *National Instruments LabView 6.1*

### 1.3 Настройка отображения кириллических шрифтов на блок-диаграмме и передней панели ВП

На сегодняшний день известна существенная проблема, препятствующая комфортной работе в *National Instruments LabView 6.1*. Связана проблема с некорректным отображением кириллических шрифтов при работе с настройками пакета прикладных программ, заданными по умолчанию. Данная проблема имеет решение.

На панели инструментов, расположенной под главным меню, необходимо нажать на самую широкую кнопку. Она предназначена для выбора и конфигурирования параметров шрифтов. Переход к основному набору настроек шрифтов производится через первый пункт выпадающего списка – «Font

*Dialog...*» (диалоговое окно настройки шрифтов). Расположение этого пункта выпадающего списка показано на Рисунке 1.3.1.

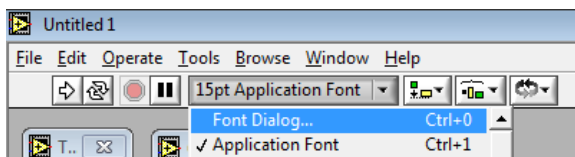


Рисунок 1.3.1 – Переход к диалоговому окну настройки шрифтов

Изменения настроек по умолчанию для всей среды разработки ВП можно добиться, если в раскрывшемся диалоговом окне выставить обе галочки «Умолчания для передней панели» («*Panel Default*») и «Умолчания для блок-диаграммы» («*Diagram Default*») одновременно (Рисунок 1.3.2). Далее в комбинированном списке шрифтов («*Font*») следует выбрать любой кириллический шрифт (для такого шрифта характерен постфикс «*Cyr*»). Все остальные умолчания подбираются на усмотрение автора-разработчика ВП, либо оставляются без изменений.

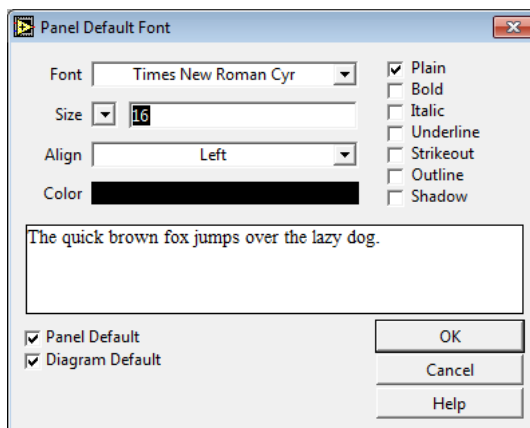


Рисунок 1.3.2 – Рекомендации по настройке шрифта, заданного в *National Instruments LabView* по умолчанию

При выставленных флагах «*Panel Default*» и «*Diagram Default*» указанные настройки шрифта будут применены по умолчанию для текущего разрабатываемого ВП, а также для последующих создаваемых ВП, но вступят в полную силу только после закрытия и повторного открытия среды.

При всём многообразии функций и возможностей пакета прикладных программ *National Instruments LabView* в каждом задании Сборника рассматривается и закрепляется только небольшой его фрагмент. Этот подход соответствует дозированному и эффективному освоению *NI LabView*.

После устранения проблем с кириллическими шрифтами необходимо проверить готовность среды разработки к нормальному и удобному функционированию.

#### 1.4 Отображение / сокрытие основных элементов управления

**Смоделируем ситуацию.** Допустим, что при первом запуске *National Instruments LabView* все панели инструментов скрыты от пользователя, как это показано на Рисунке 1.4.1.

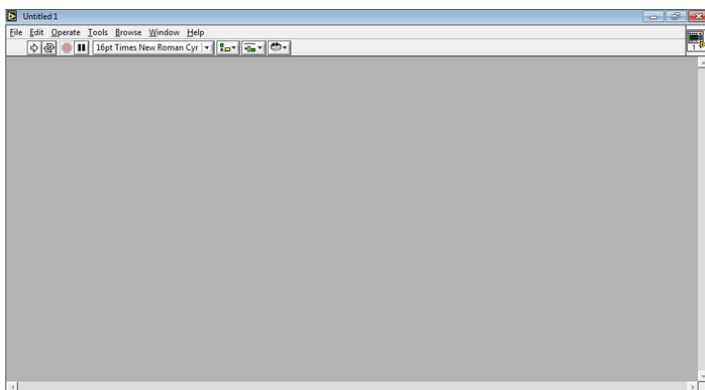


Рисунок 1.4.1 – Состояние *National Instruments LabView 6.1*, при котором основные панели инструментов скрыты

Для активации вывода на рабочей области основных панелей инструментов достаточно помнить, какие из пунктов меню связаны с отображением / сокрытием этих панелей инструментов. Они находятся в пункте меню «*Window*».

Сначала активируем вывод на экран панели интерфейсных элементов управления (они позволяют обеспечить взаимодействие конечного пользователя / разработчика ВП с внутренними механизмами ВП). Для этого в главном меню следует пройти путь «*Window* > *Show Controls Palette*» как это показано на Рисунке 1.4.2.

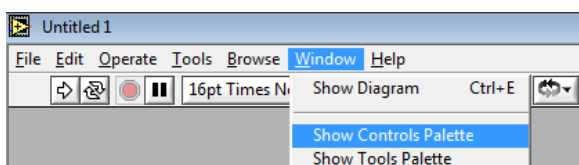


Рисунок 1.4.2 – Переход по меню к отображению панели интерфейсных элементов управления

В результате выбора указанного пункта меню над передней панелью ВП отобразится структура «*Controls*», представленная на Рисунке 1.4.3, или аналогичная ей.

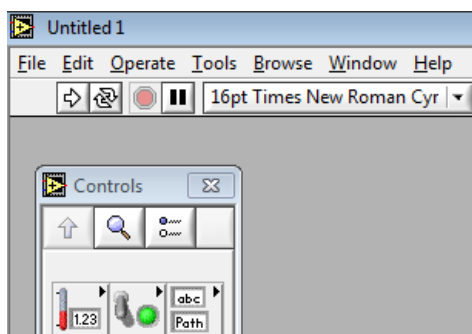


Рисунок 1.4.3 – Отображение панели интерфейсных элементов управления («*Controls*»)

Далее отобразим панель основных инструментов, предоставляющих возможность пользователю / разработчику ВП манипулировать элементами интерфейса, а также элементами в составе графического кода. Для этого в главном меню необходимо пройти путь «*Window > Show Tools Palette*», как показано на Рисунке 1.4.4.

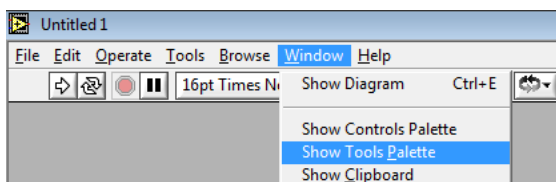


Рисунок 1.4.4 – Переход по меню к отображению панели основных инструментов

В результате выбора указанного пункта меню над передней панелью ВП отобразится структура «*Tools*», представленная на Рисунке 1.4.5.

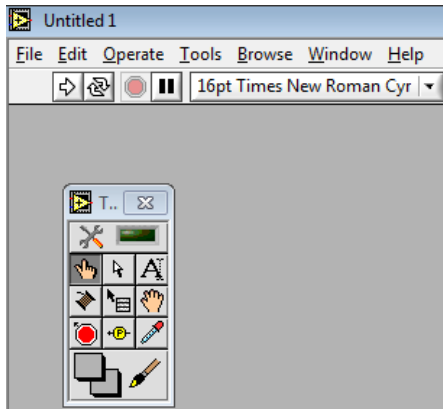


Рисунок 1.4.5 – Отображение основных инструментов («*Tools*»)

Панель основных инструментов является общей как для передней панели ВП, так и для блок-диаграммы ВП.



К настоящему моменту без рассмотрения осталась только одна панель из числа часто используемых – панель функций среды разработки. Она доступна для вызова только в режиме составления блок-диаграммы ВП. В главном меню необходимо пройти путь «*Window > Show Functions Palette*» (Рисунок 1.4.6).

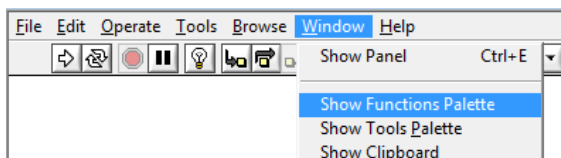


Рисунок 1.4.6 – Переход по меню к отображению панели функций среды

В результате выбора указанного пункта меню над блок-диаграммой ВП отобразится структура «*Functions*», представленная на Рисунке 1.4.7.

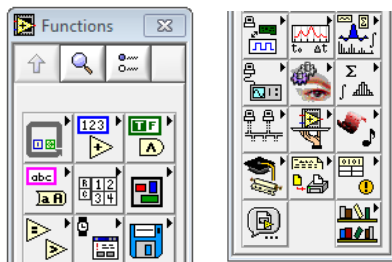


Рисунок 1.4.7 – Отображение панели функций среды («*Functions*»)

После вызова всех упомянутых панелей считается, что пакет прикладных программ *National Instruments LabView* готов к нормальному взаимодействию в направлении разработки нового ВП.

## 1.5 Размещение интерфейсных элементов управления на передней панели ВП

Приступим к размещению интерфейсных элементов управления на передней панели ВП. Окно передней панели должно быть активно (Рисунке 1.5.1).

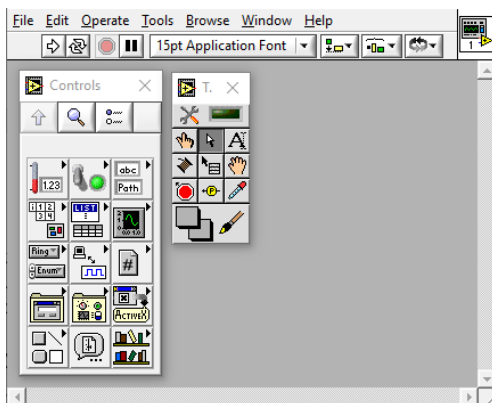


Рисунок 1.5.1 – Передняя панель ВП, подготовленная к работе

В перечне доступных интерфейсных элементов управления необходимо перейти в раздел «Числовых» («*Numeric*»), как показано на Рисунке 1.5.2.

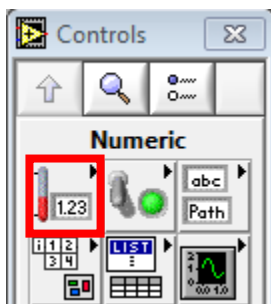


Рисунок 1.5.2 – Отметка в панели интерфейсных элементов управления («*Controls*») раздела числовых элементов («*Numeric*»)

В раскрывшемся перечне выбрать «Числовой элемент ввода» («*Digital Control*») (Рисунок 1.5.3), необходимый для ввода каких-либо значений оператором в программу, написанную на графическом языке программирования.

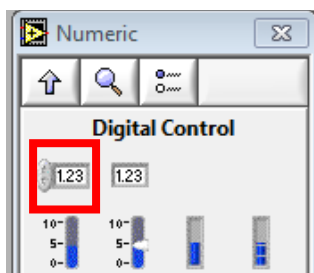


Рисунок 1.5.3 – Содержимое раздела числовых элементов управления («*Numeric*») с подсветкой числового элемента ввода («*Digital Control*»)

Разместим выбранный числовой элемент ввода на передней панели ВП (Рисунок 1.5.4). После выбора элемента из перечня под курсором мыши появляется его пунктирный «фантом», переместив который в интересующую область передней панели достаточно однократно щёлкнуть мышью для фиксации элемента в указанной позиции. Аналогичным способом на передней панели может быть размещено несколько таких же числовых элементов ввода. Одновременно с размещением на панели образа элемента, на блок-диаграмме размечается его представление в виде графической переменной (это подробнее показано в разделах 1.7 и 1.8).

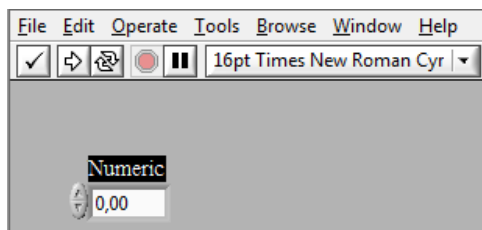


Рисунок 1.5.4 – Размещение эл-та ввода на передней панели ВП

На передней панели ВП раздел числовых элементов управления остался активен. Выберем в нём соседний элемент управления – числовой элемент вывода (Рисунок 1.5.5), необходимый для вывода / индикации результирующей или промежуточной расчётной информации на переднюю панель.

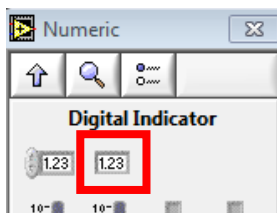


Рисунок 1.5.5 – Содержимое раздела числовых элементов («*Numeric*») с подсветкой числового элемента вывода информации («*Digital Indicator*»)

Разместим выбранный числовой элемент вывода на передней панели ВП рядом с ранее размещённым элементом ввода (Рисунок 1.5.6).

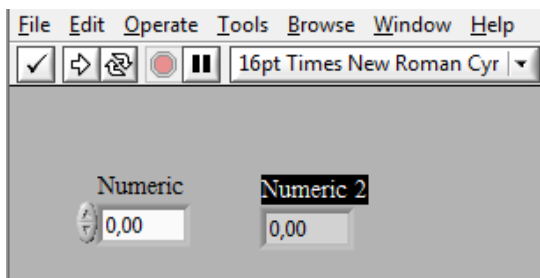


Рисунок 1.5.6 – Размещение числового элемента вывода («*Digital Indicator*») на передней панели ВП

После щелчка мышью в свободной области передней панели ВП снимаем выделение / маркировку с последнего размещённого на графическом пользовательском интерфейсе компонента и в этот момент считаем заготовку / шаблон для ВП «Вход-Выход» выполненной.

Согласно одному из правил вежливости программиста, переменным необходимо давать значащие имена, которые бы позволили лицу, намеренному или поставленному продолжить написание ранее составленного кода, быстрее разобраться в нём.

В *NI LabView* имена переменных однозначно связаны с именами элементов на передней панели ВП, потому проявляя заботу о конечном пользователе, одновременно получается заботиться и о программисте. Для выдачи значащих имён интерфейсным элементам управления используется инструмент редактирования текста (Рисунок 1.5.7).

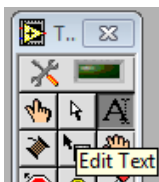


Рисунок 1.5.7 – Инструмент для редактирования текста («Edit Text»)

Программный продукт, не дружелюбный для пользователя, является непригодным для использования.

Итого, руководствуясь вышесказанным, назовём элемент ввода «Входом», а элемент вывода – «Выходом» (можно назвать синонимично «Вводом» и «Выводом», соответственно). Результат изменения имён интерфейсных элементов управления показан на Рисунке 1.5.8.

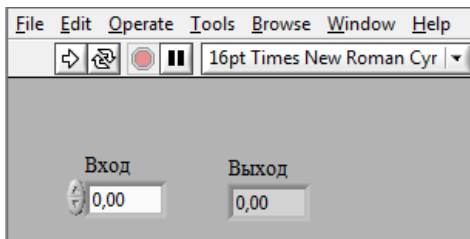


Рисунок 1.5.8 – Присвоение значащих имён переменным, связанным с элементами ввода и вывода

Если с эстетической точки зрения начинает казаться, что элементы расположены неровно, то можно произвести их выравнивание относительно друг друга, предварительно выделив их и выбрав одну из операций выравнивания, как это показано на Рисунке 1.5.9.

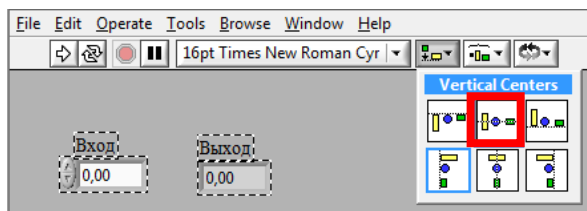


Рисунок 1.5.9 – Выравнивание элементов ввода и вывода относительно друг друга по общей для них центральной и горизонтальной оси

## 1.6 Размещение функций на блок-диаграмме ВП

Размещение функций на блок-диаграмме ВП реализуется при активном окне блок-диаграммы (Рисунке 1.6.1).

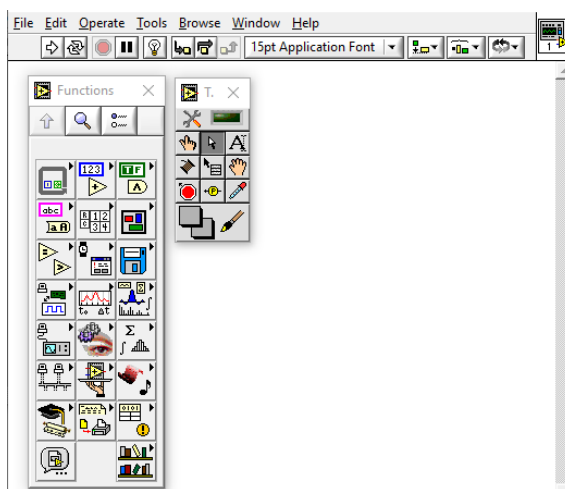



Рисунок 1.6.1 – Активное окно блок-диаграммы ВП

The screenshot shows the 'Functions' palette with the 'Numeric' category selected. The 'Add' function is highlighted, and its icon is visible in the main workspace area.

число элементов, размещаемых на блок-диаграмме



На Рисунке 1.6.4 представлена краткая информация об операторе сложения. Этот оператор – бинарный, он требует подачи на вход (подключения к клеммам, расположенным слева) значений от двух операндов с именами «x» и «y». Результат (выходная информация), равный «x+y» при подключении клеммы, расположенной справа, например, к числовому элементу вывода, передаётся и отображается на нём.

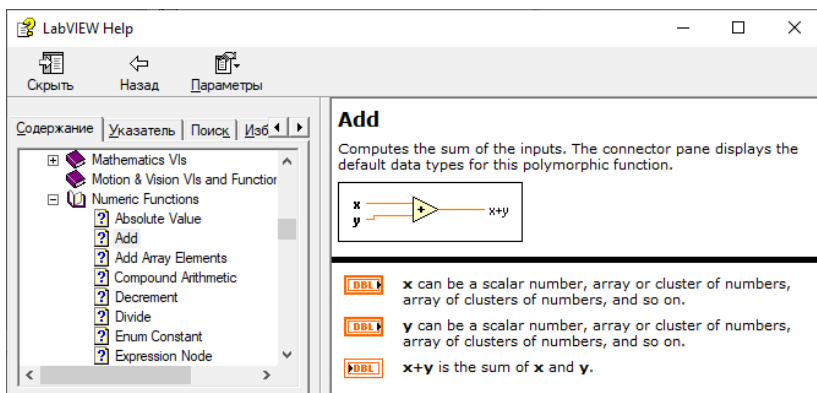


Рисунок 1.6.4 – Справка об операции сложения

Подытожим. Операция сложения содержит всего три клеммы: две входных для операндов (слева) и одну выходную для результата справа. Поэтапное подключение клемм операции сложения показано на Рисунке 1.6.5. Более подробно аналогичные соединения рассмотрены в разделе 1.7.

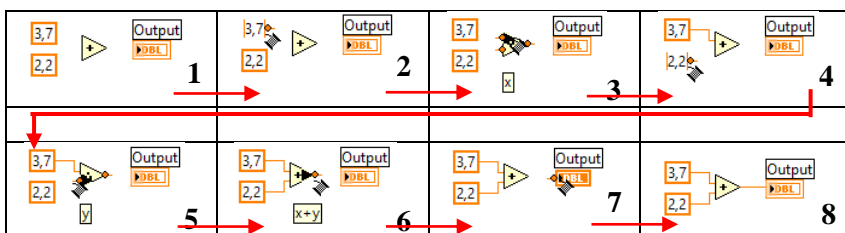


Рисунок 1.6.5 – Подключение операции сложения к константам и числовому элементу вывода

Аналогичную справку можно получить, в частности, по операции деления (Рисунок 1.6.6). Операция деления является одной из тех операций, в которых важен порядок подключения операндов. Видно, что слева через «x» обозначена входная клемма делимого, через «y» – входная клемма делителя, а справа через «x/y» обозначена выходная клемма частного (результата деления).



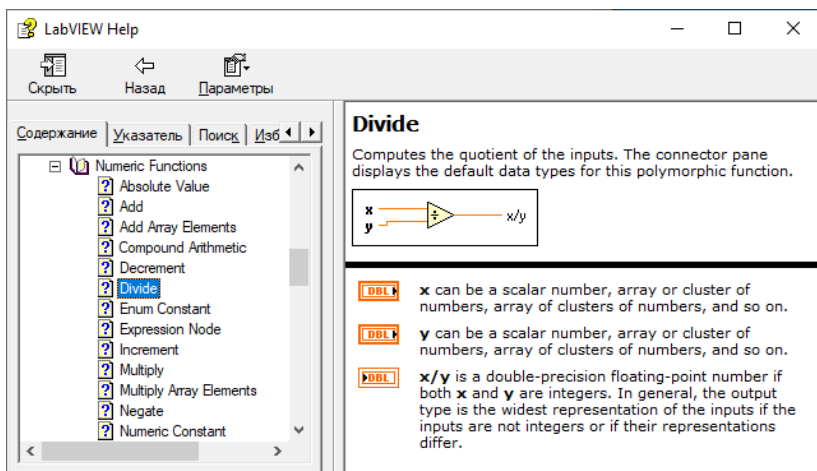


Рисунок 1.6.6 – Справка по операции деления

Более сложные и не часто встречающиеся в расчётах операции снабжены развёрнутой (детализированной) справочной информацией (Рисунок 1.6.7), в отдельных случаях, с указанием расчётных формул, закреплённых за результирующими клеммами. Детализированная структура справочной информации показана на примере операции, объединяющей в себе результат целочисленного деления «*x*» на «*y*» «*Quotient*» («Частное»), кратко обозначенный как «*IQ*», и остаток от целочисленного деления «*x*» на «*y*» «*Remainder*» («Остаток»).

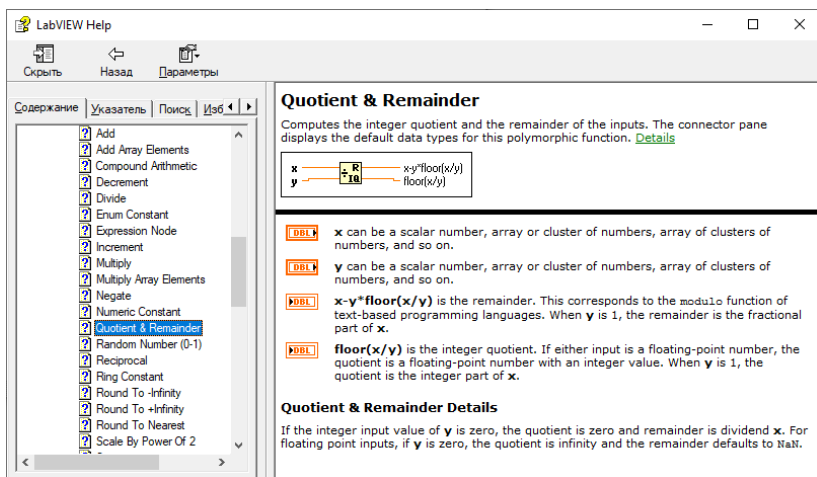


Рисунок 1.6.7 – Справка по операции целочисленного деления с остатком

При работе с *NI LabView* важно научиться внимательно читать информацию, вынесенную разработчиками в качестве наименований некоторых функций, – в ряде случаев она уже является развёрнутой подсказкой о работе функций.

Так, например, датчик псевдослучайных чисел (Рисунок 1.6.8), равномерно распределённых от 0 до 1 («*Random Number 0-1*»), содержит довольно краткое и даже избыточное описание. Избыточность состоит в том, что сведения о распределении реализаций от 0 до 1 содержится в заголовке.

Функция достаточно проста в использовании. У неё нет входных клемм, а значит, и входных параметров. Есть только одна выходная клемма – функция генерирует псевдослучайные значения.

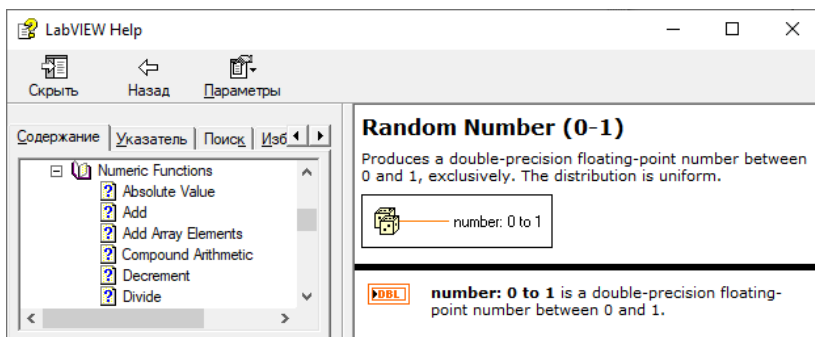


Рисунок 1.6.8 – Справка по датчику псевдослучайных чисел, распределённых равномерно от 0 до 1

Следует так же внимательно присмотреться к обозначениям типов данных и типов линий (связей). Научившись работать с информацией двух упомянутых категорий, получится несложно и быстро освоить стандартные оптимизационные функции и вычислители, которые необходимо просто правильно подключить для получения высокоэффективных и точных результатов.

Типам данных и типам связей посвящён раздел 1.9. Одной из числовых функций, наглядно демонстрирующей различие типов связей и данных является функция суммирования всех элементов массива («*Add Array Elements*»). Эта функция рассмотрена на Рисунке 1.6.9.

Видно, что слева к ней подходит более жирная линия (одномерный массив), по сравнению с линией, выходящей справа (сумма элементов массива – это одно единственное, конкретное значение; в общем случае – вещественное значение).

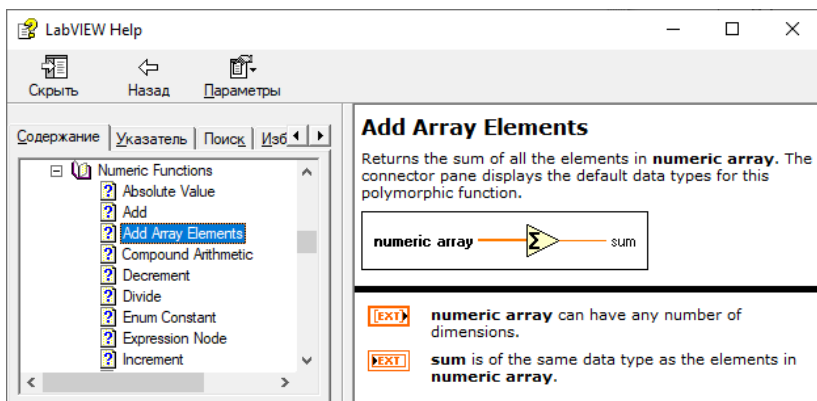


Рисунок 1.6.9 – Справка по суммированию элементов массива

Отдельная группа операторов содержит в разделе справки и полезные подсказки по их использованию (Рисунок 1.6.10). Так, например, справочный раздел оператора выбора («*Select*») делает отсылку к структуре выбора («*Case Structure*») для ситуаций, в которых требуется работа с наборами (массивами) значений, а не с отдельными значениями, как в разобранный на иллюстрации случае использования оператора выбора.

Задание №11, сформулированное в разделе 4 настоящего Сборника типовых задач, ориентировано на то, что разработчик примет правильное, взвешенное решение и воспользуется одним из предложенных инструментов разветвления вычислительного процесса.

Работа оператора выбора сводится к тому, что если некоторый результат расчёта логического выражения, подключенный и переданный ко входной клемме «*s*» оказался истинным, то на выходе оператора окажется то значение, которое подключено к верхней входной клемме «*t*». Если результат расчёта логического выражения оказался ложным, то на выходе оператора окажется то значение, которое подведено к его нижней входной клемме «*f*».

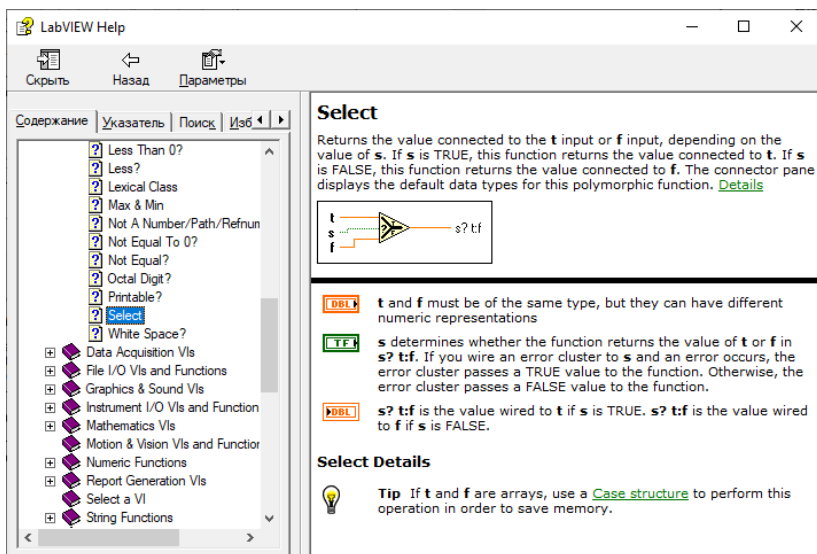


Рисунок 1.6.10 – Справка по оператору выбора одного из пары поступающих на его вход значений в зависимости от результата вычисления логического выражения так же поступающего на его вход

## 1.7 Соединение образов интерфейсных элементов управления на блок-диаграмме ВП

Нет ничего проще в *NI LabView*, чем задать однозначное соответствие между входными и выходными данными одного и того же типа. Решим эту задачу для числовых («*Numeric*») элементов управления.

После приведения в порядок содержимого лицевой панели ВП, продемонстрированного в разделе 1.5, переходим к кодированию, – на блок-диаграмму ВП (Рисунок 1.7.1).

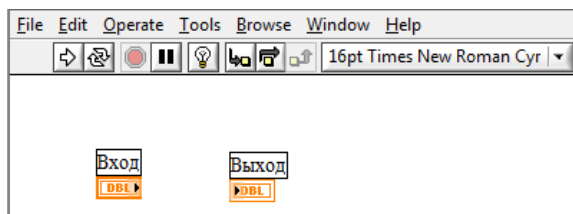


Рисунок 1.7.1 – Отображение ранее созданных элементов ввода и вывода на блок-диаграмме ВП

Видно, что графические переменные однозначно соответствуют графическим элементам управления, размещённым ранее на передней панели ВП.

Следует отметить, что чтение и обработка информации происходит слева направо, потому важно следить за тем, чтобы элементы ввода информации / поставщики исходных данных располагались левее обработчиков данных (операций), а также элементов вывода информации / приёмников результатов.

Для элементов ввода информации характерно наличие чёрной стрелки, направленной вправо и расположенной у правой границы элемента (Рисунок 1.7.2). Это выходная клемма, предполагающая подключение к ней других элементов посредством соединительного провода для передачи им введённого через переднюю панель значения.

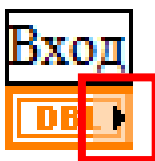


Рисунок 1.7.2 – Отметка выходной клеммы образа интерфейсного элемента для ввода данных

Для элементов вывода информации характерно наличие чёрной стрелки, направленной вправо и расположенной у левой границы элемента (Рисунок 1.7.3). Это входная клемма, предполагающая подключение к ней других элементов

посредством соединительного провода для передачи от них результата, который необходимо отобразить на передней панели ВП.



Рисунок 1.7.3 – Отметка входной клеммы образа интерфейсного элемента для вывода данных

Решим поставленную простую задачу ввода / вывода – соединим переменные между собой, для чего выберем соединительный провод («*Connect Wire*») (Рисунок 1.7.4) и последовательно нажмём левой кнопкой мыши на мигающей правой границе «Входа» (Рисунок 1.7.5) и затем на мигающей левой границе «Выхода» (Рисунок 1.7.6).



Рисунок 1.7.4 – Выбор инструмента «Соединительный Провод» («*Connect Wire*»)

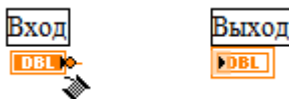


Рисунок 1.7.5 – Подключение начала соединительного провода к выходной клемме «Входа»

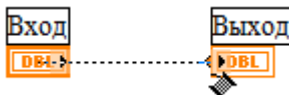


Рисунок 1.7.6 – Подключение конца соединительного провода к входной клемме «Выхода»

Результат соединения «Входа» с «Выходом» можно наблюдать на Рисунке 1.7.7.

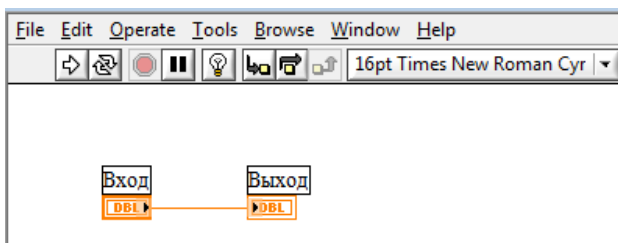


Рисунок 1.7.7 – Результат соединения элементов ввода и вывода

Для обеспечения возможности изменения значения на элементе ввода, сменим соединительный провод на инструмент изменения значений («*Operate Value*»), выглядящий как указательный палец правой руки (Рисунок 1.7.8).



Рисунок 1.7.8 – Выбор инструмента изменения значений элементов («*Operate Value*»)

При использовании данного инструмента увеличим значение на входе до «5,00», как показано на Рисунке 1.7.9 и запустим ВП в режиме однократного исполнения.



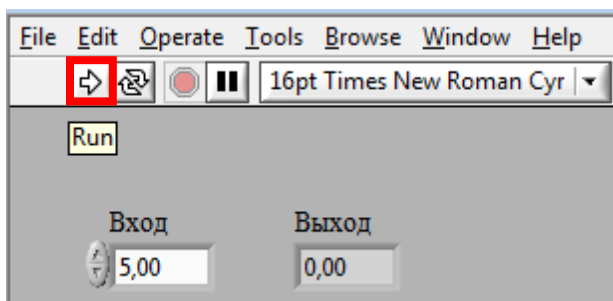


Рисунок 1.7.9 – Изменение значения, хранимого на элементе ввода и переход к разовому запуску ВП на исполнение («Run»)

Видно (Рисунок 1.7.10), что результатом работы ВП становится настройка соответствия «Выхода» со «Входом».

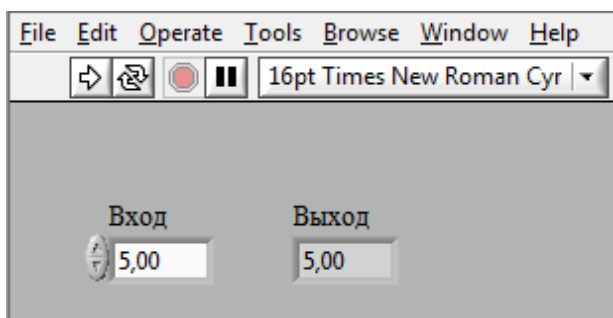


Рисунок 1.7.10 – Результат разового запуска ВП на исполнение

Обучающимся предлагается попробовать проделать самостоятельно серию экспериментов, изменяя значения на «Входе» и запуская программу в режиме однократного исполнения («Run»), а также запустив программу в режиме циклического исполнения («Run Continuously») и меняя значения на входе.

## 1.8 Интерпретация связи интерфейсных элементов управления с блок-диаграммой ВП

На Рисунке 1.8.1 показана интерпретация типовой задачи ввода-вывода, которая увязывает между собой оператора, переднюю панель ВП и блок-диаграмму ВП.

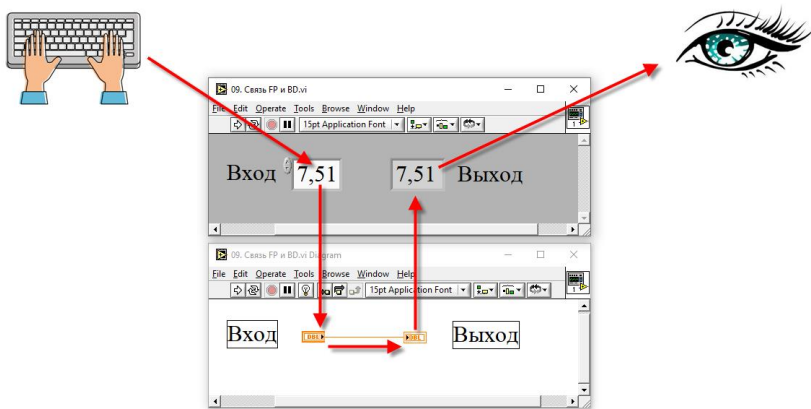


Рисунок 1.8.1 – Интерпретация типовой задачи ввода-вывода

## 1.9 Типы значений и типы связей

До настоящего момента все рассмотренные элементы на блок-диаграмме были окрашены, в основном, в оранжевый цвет. Оранжевый цвет в *National Instruments LabView* характерен для вещественных значений.

Ещё одно важное правило вежливости программиста гласит: необходимо составлять программы в общем виде. В частности, здесь подразумевается и тот факт, что под взаимодействие с пользователем должны выделяться обобщающие типы данных (под которые выделятся больше всего памяти). Среди множества числовых типов данных в качестве обобщающего принимается вещественный тип двойной точности (*double / DBL*).

По умолчанию это правило соблюдается в *NI LabView*. Но что делать, когда необходимы, например, только целые числовые значения?

На Рисунке 1.9.1 показано как через контекст числового элемента вывода перейти от выводимых вещественных значений двойной точности к 16-тиразрядным целым значениям. Последовательно пройти в пункт «*Representation*» («Представление») и в нём выбрать «*Word*» («Слово»), как и отмечено курсором.

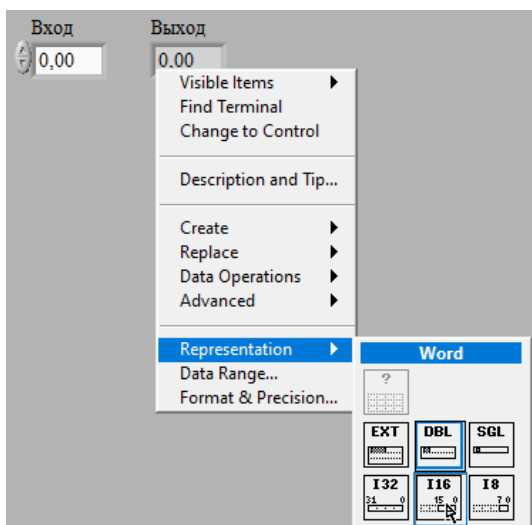


Рисунок 1.9.1 – Перевод числового элемента вывода с именем «Выход» из режима отображения вещественных значений в режим отображения целых значений

Произошедшие изменения можно мгновенно отследить как по лицевой панели ВП, так и по блок-диаграмме. На лицевой панели будет отображаться в точности ноль («0»), а на блок-диаграмме образ элемента вывода, названного «Выход», окрасится в синий цвет. Обнаруживая на блок-диаграмме компоненты синего цвета, можно с уверенностью говорить о том, что они – целочисленные.

При этом *NI LabView* не входит в перечень «капризных» сред разработки программного обеспечения. Это означает, что на целочисленные элементы можно подавать, в частности, и вещественные значения (как показано на Рисунке 1.9.2). «Капризные» / строгие среды разработки программного обеспечения не позволили бы элементами более мощного множества переписывать элементы менее мощного множества. В проиллюстрированной же ситуации важно понимать, что передаваемый вещественный результат будет округлѐн до ближайшего целого значения.

Запись же целых значений в вещественные переменные – это абсолютно нормальная практика: элементы менее мощного множества могут перезаписывать элементы более мощного множества.

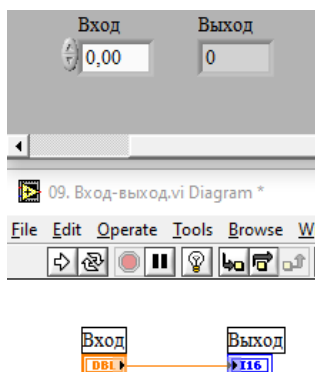


Рисунок 1.9.2 – Пример записи элемента из более мощного множества поверх элемента из менее мощного множества (некорректная ситуация)

На Рисунке 1.9.3 показано соответствие типов, когда и входная и выходная информация одного – целого типа.

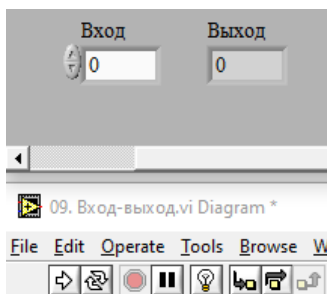


Рисунок 1.9.3 – Пример соответствия типов при «вводе-выводе»: «целое-целое»

На Рисунке 1.9.4 разобрана оставшаяся нерассмотренной (корректная) ситуация, когда целое значение записывается в вещественный контейнер (вещественную переменную).

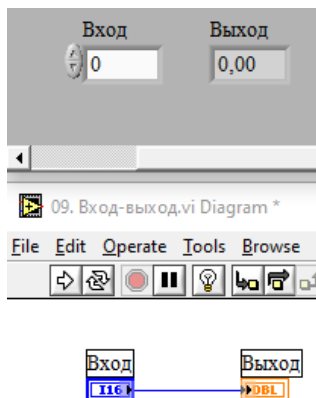


Рисунок 1.9.4 – Пример записи элемента из менее мощного множества поверх элемента из более мощного множества (корректная ситуация)

Опять же, до настоящего момента работа проводилась только внутри разделов числовых функций и числовых интерфейсных элементов управления. Для более углубленного изучения *NI LabView* при переходе от линейного вычислительного процесса к разветвляющемуся необходимо освоить логические (Булевские) элементы, размещаемые как на лицевой панели, так и на блок-диаграмме. На лицевой панели – это всевозможные тумблеры и лампочки, на блок-диаграмме – это образы элементов и элементы зелёного цвета. Они находятся в одном из двух состояний «истина» («да» / «включено» / «логическая единица» / «+»), либо «ложь» («нет» / «выключено» / «логический ноль» / «-»).

На Рисунке 1.9.5 показано добавление логических элементов в ВП и решение для них типовой задачи «ввода-вывода».

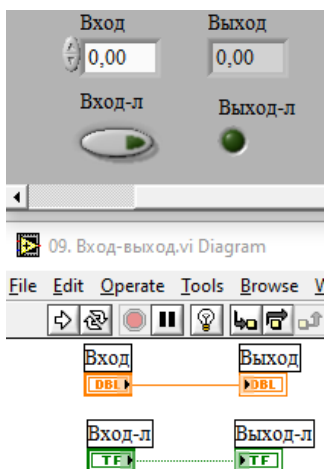


Рисунок 1.9.5 – Добавление в ВП логических элементов и решение для них типовой задачи «ввода-вывода»

Далее на Рисунке 1.9.6 продемонстрирована попытка связать между собой элементы, принципиально не согласующиеся друг с другом по типу. Линии связи при этом становятся пунктирными.

Тип соединения, обозначенный пунктиром в *NI LabView*, носит наименование «неактивного соединения». Иные типы соединений, например, все рассмотренные ранее, – называют «активным соединением».

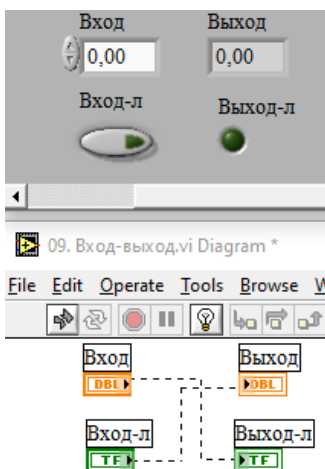


Рисунок 1.9.6 – Пример несоответствия типов для связанных образов интерфейсных элементов в *National Instruments LabView*

Для устранения всех неактивных соединений в *NI LabView* существует удобная комбинация горячих клавиш: убрать все оборванные связи можно по сочетанию «*Ctrl*» + «*B*».

Это бывает особенно полезно, когда оборванные связи не видно за пределами обозреваемой части блок-диаграммы или когда оборванные связи оказались под структурами и их так же не видно.

Толщина активных соединений определяет мощность используемых множеств данных. Для наглядной демонстрации различных значений толщины линий связи необходимо ввести, например, массив.

Не типизированный (пустой) массив находится в соответствующем разделе массивов и кластеров («*Array &*

*Cluster*)). Элемент «Array», размещаемый на лицевой панели ВП, показан на Рисунке 1.9.7.

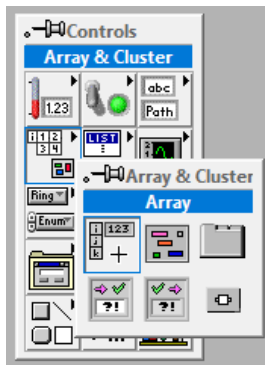


Рисунок 1.9.7 – Место массива («Array») в перечне интерфейсных элементов управления

Массив, тип которого не определён, на блок-диаграмме отмечается образом чёрного цвета как неактивный интерфейсный элемент управления (Рисунок 1.9.8).

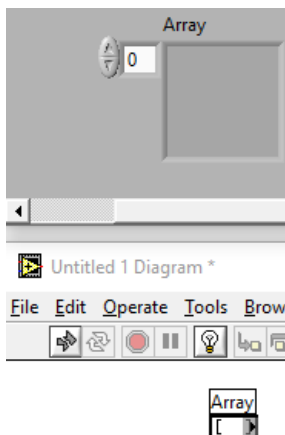


Рисунок 1.9.8 – Пример размещения не типизированного массива



Если же в перечне интерфейсных элементов управления выбрать, например, числовой элемент ввода информации и перетащить его в область массива, то сам массив получит свой вещественный тип в соответствии с настройками по умолчанию элемента ввода и, станет, собственно, массивом исходных данных (Рисунок 1.9.9).

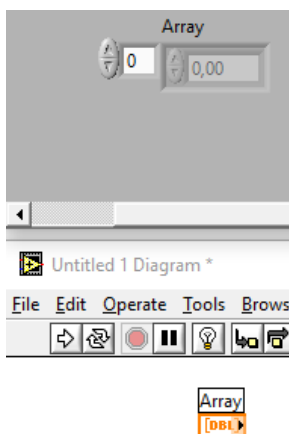


Рисунок 1.9.9 – Пример размещения одномерного вещественного массива исходных сведений

Этот массив теперь можно подключать к другим элементам и/или операциям. Для удобства демонстрации и соблюдения логики повествования возьмём операцию, рассмотренную по справочнику в пункте 1.6 – операцию суммирования элементов массива. При подключении к ней заметна яркая оранжевая линия (Рисунок 1.9.10), которая не противоречит входному типу информации, требующейся операции согласно её формату.

Для наглядности сопоставления толщины линий необходимо доукомплектовать операцию связями (не только слева, но и справа).

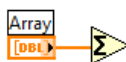
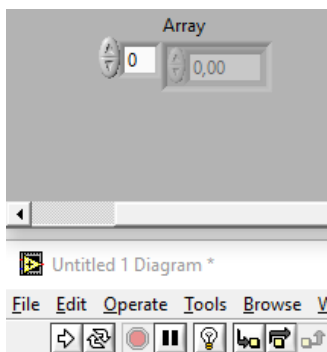


Рисунок 1.9.10 – Пример подключения исходного вещественного массива к операции суммирования элементов массива

Добавим на лицевую панель ВП числовой элемент вывода и соединим его с выходом операции поиска суммы элементов массива (Рисунок 1.9.11).

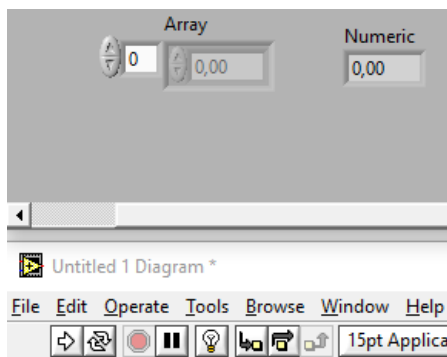


Рисунок 1.9.11 – Пример использования операции суммирования элементов одномерного массива

Проверим, связано ли увеличение мощностей используемых множеств с пропорциональным увеличением толщины линий связи? – Добавим размерность через контекст как показано на Рисунке 1.9.12 – сделаем из вектора матрицу.

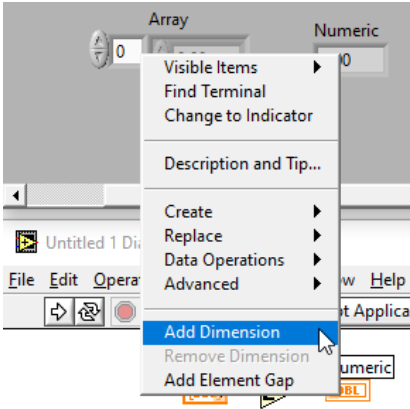


Рисунок 1.9.12 – Добавление размерности массиву (трансформация вектора в матрицу)

Как результат – толщина линии связи увеличилась (Рисунок 1.9.13).

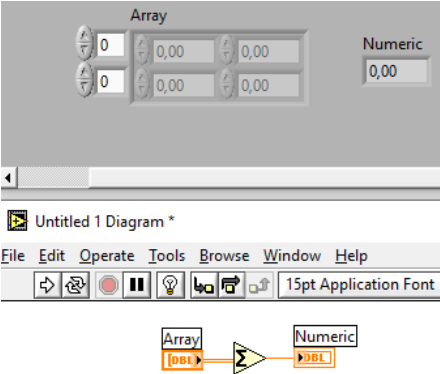


Рисунок 1.9.13 – Пример использования операции суммирования элементов двумерного массива

Сделаем очередной исследовательский шаг – добавим массиву ещё одну размерность: перейдём от двумерного массива к трёхмерному. Толщина линии связи снова увеличилась (Рисунок 1.9.14).

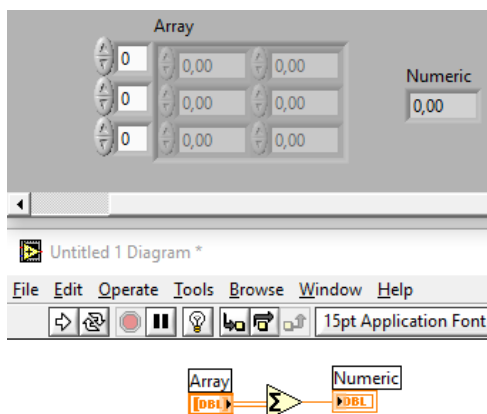


Рисунок 1.9.14 – Пример использования операции суммирования элементов трёхмерного массива

Другим известным примером более толстой линии связи является тип данных «строка» («string»), обозначаемый фиолетовым (розовым) цветом. Это не случайно. Если задаться целью и прочитать строгое определение строки в программировании, то в большинстве источников строка будет определена как «массив символов» (Рисунок 1.9.15).

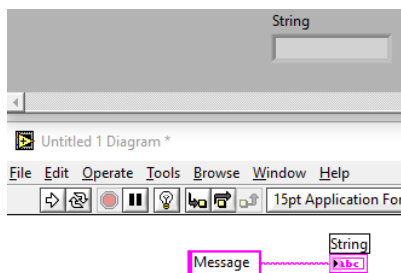


Рисунок 1.9.15 – Пример связи элементов строкового типа

Та же строка, но предназначенная для достижения иных целей и задач, в *NI LabView* существует в формате дополнительного типа данных, называемого «путём к файлу» («*File Path*»).

Пути к файлам на блок-диаграмме обладают характерной бирюзовой окраской. Они, так же как и строки, имеют чуть большую толщину. Они тоже являются массивами символов.

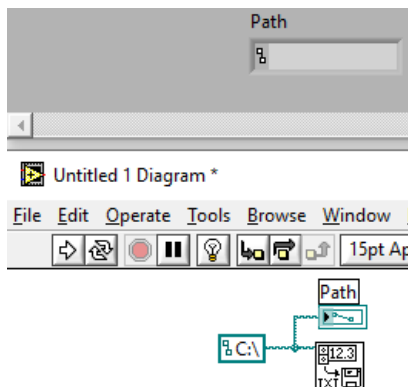


Рисунок 1.9.16 – Пример связи элементов типа «путь к файлу»

## 2 Задание «Разработка ВП "Простой калькулятор"»

В пакете прикладных программ *National Instruments LabView* разработать ВП «Простой Калькулятор», позволяющий оператору по итогам работы этого ВП наблюдать результаты выполнения следующих арифметических операций:

- сложение двух операндов,
- вычитание одного операнда из другого,
- умножение двух операндов,
- деление одного операнда на другой (с выводом на интерфейс ошибки при делении на ноль и на близкие к нулю значения).

В рамках разработки графического пользовательского интерфейса предусмотреть:

- две ячейки («*Digital Control*») для ввода числовой информации (по количеству операндов, над которыми выполняются арифметические операции);
- четыре ячейки («*Digital Indicator*») для вывода числовой информации (по количеству используемых арифметических операций);
- лампочку («*Boolean Indicator*» / «*Boolean LED*») для вывода логической информации с целью реагирования на ошибочный ввод исходных данных – ноль (или близкое к нулю) в знаменателе.

Все интерфейсные элементы управления должны содержать значащие имена, отражающие назначение этих элементов, например, ячейки для ввода информации могут быть названы:

- «Операнд 1», «Операнд 2»;
- «Слагаемое 1 / Уменьшаемое / Множитель 1 / Делимое», «Слагаемое 2 / Вычитаемое / Множитель 2 / Делитель».

Необходимо продумать индивидуальный авторский дизайн графического пользовательского интерфейса ВП «Простой Калькулятор». Двух одинаковых дизайнерских решений для ВП «Простой Калькулятор» в одной учебной студенческой группе быть не должно.

Изменения, вносимые в исходные данные, должны сказываться на отображаемых результатах в реальном времени.

При подготовке блок-диаграммы графического кода программы минимизировать количество пересечений линий связи, а также попытаться создать максимально компактную и читаемую схему.

Выше сформулированы требования для выполнения только общей части работы. Варианты индивидуальных задач и требования к ним сформулированы в параграфе 2.4.

По итогам выполнения работы на проверку по электронной почте сдаются строго четыре файла (файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3):

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (\*.doc или \*.docx) с учётом рекомендаций, собранных в Приложении 2;
- файл ВП *National Instruments LabView* (\*.vi) по общей части задания;
- файл ВП *National Instruments LabView* (\*.vi) по индивидуальной части задания, связанной с закреплением навыков использования переменных (числовых элементов ввода и числовых элементов вывода);
- файл ВП *National Instruments LabView* (\*.vi) по индивидуальной части задания, связанной с закреплением навыков использования констант и некоторых стандартных функций.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

В программе «Учебной практики» заданию присвоен номер «9».

## **2.1 Цель работы**

Закрепление базовых навыков работы в пакете прикладных программ *National Instruments LabView*, развитие навыков алгоритмизации и графического программирования. Изучение раздела числовых («*Numeric*») и логических («*Boolean*») элементов управления, раздела функций сравнения («*Comparison*»).

## **2.2 Полезные соотношения для выполнения задания**

### **2.2.1 Термины**

Операнд в языках программирования – аргумент операции; данные, которые обрабатываются командой; грамматическая конструкция, обозначающая выражение, задающее значение аргумента операции [12].

Операция – конструкция в языках программирования, аналогичная по записи математическим операциям, то есть специальный способ записи некоторых действий [13].

### **2.2.2 Связь между числовым и логическим элементами управления ВП**

Рассматривается решение другой простой задачи, в которой проверяется наличие соответствия между входной информацией и некоторой наперёд заданной константой (в данном примере её значение равно «5»). Ответ, получаемый в результате сопоставления значений, – логического типа.

Все логические интерфейсные элементы управления содержатся в разделе логических или булевских («*Boolean*») (Рисунок 2.2.2.1).



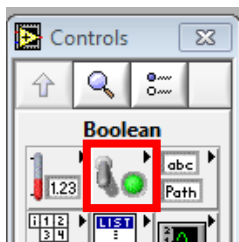


Рисунок 2.2.2.1 – Подсветка в панели интерфейсных элементов управления раздела логических элементов («*Boolean*»)

В перечне логических элементов управления существуют два наиболее примечательных светодиодных элемента вывода («*LED*»):

- «квадратный» (это только название, а на самом деле – элемент прямоугольной формы) светодиод («*Square LED*»);
- «круглый» светодиод («*Round LED*»).

По смысловой нагрузке эти элементы равнозначны, потому размещаются на графическом пользовательском интерфейсе по усмотрению / по желанию автора ВП (Рисунок 2.2.2.2).

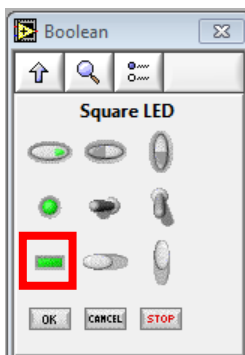


Рисунок 2.2.2.2 – Содержимое раздела логических элементов («*Boolean*») с подсветкой логического элемента вывода типа «прямоугольный» светодиод («*Square LED*»)

Как правило, круглые элементы управления выглядят менее строго и не столь же быстро привлекают к себе внимание оператора, как, например, прямоугольные элементы (Рисунок 2.2.2.3).

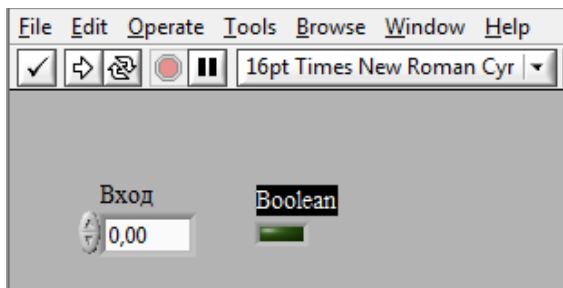


Рисунок 2.2.2.3 – Размещение «прямоугольного» светодиода («Square LED») на передней панели ВП

Здесь так же не стоит забывать о существовании правил вежливости программиста и общих принципов обеспечения дружелюбности графического пользовательского интерфейса. Новому логическому элементу вывода следует присвоить одно из имён, соответствующих выходным данным: «Ответ» / «Результат» / «Отклик» / «Выход» (Рисунок 2.2.2.4).

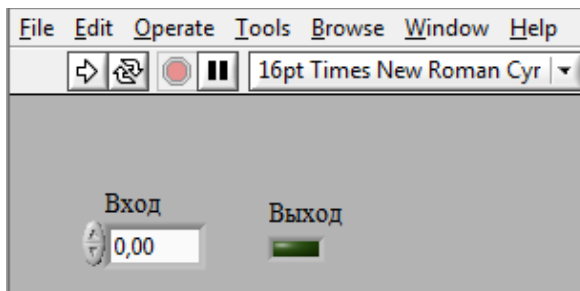


Рисунок 2.2.2.4 – Формирование значащего имени («Выход») для переменной, связанной с логическим «прямоугольным» светодиодным элементом вывода

Дальнейшая работа по настройке ВП проводится на блок-диаграмме. Для размещения константы, с которой будет выполняться сопоставление входного сигнала, следует обратиться к разделу числовых элементов управления (Рисунок 2.2.2.5).

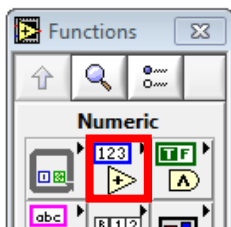


Рисунок 2.2.2.5 – Подсветка в панели функций раздела числовых элементов («*Numeric*»)

Из раздела понадобится числовая константа («*Numeric Constant*»), расположенная в левом нижнем углу окна функций («*Functions*») (Рисунок 2.2.2.6).

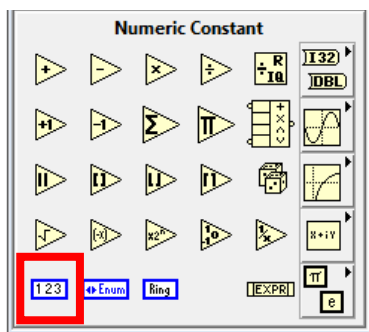


Рисунок 2.2.2.6 – Содержимое раздела числовых функций («*Numeric*») с подсветкой числовой константы («*Numeric Constant*»)

Любая **константа** в программировании характеризуется типом и значением, причём значение константы определяет её тип. Таким образом, пока в рамке указано целое значение – это будет целочисленная константа («*Integer Constant*») (Рисунок 2.2.2.7), заключённая в рамку синего цвета. Как только пользователь изменит значение константы на какое-либо, содержащее десятичный разделитель (в *NI LabView* это – запятая), то на блок-диаграмме ВП будет обозначена уже вещественная константа («*Real Constant*» / «*Single Constant*» / «*Double Constant*»), заключённая в рамку оранжевого цвета.

**Это полезно:** в рамку константы оранжевого цвета существует возможность записать строковое значение «*NaN*» («*Not a Number*» – не числовое значение). Константа «не числового» значения в ряде случаев позволяет реализовывать удобный отлов ошибок.



Рисунок 2.2.2.7 – Размещение целочисленной константы («*Numeric Constant*») на блок-диаграмме ВП

В примере не рассматривается выход за область (диапазон) целых чисел. В качестве константы на блок-диаграмме ВП выставлено значение равное «5» (Рисунок 2.2.2.8).



5

Рисунок 2.2.2.8 – Изменение значения целочисленной константы с «0» на «5»

После этого следует перейти к сравнению значения переменной, задаваемого числовым элементом ввода, с константой, для чего обратиться в панели функций («*Functions*») области блок-диаграммы ВП к разделу элементов сравнения («*Comparison*») (Рисунок 2.2.2.9).

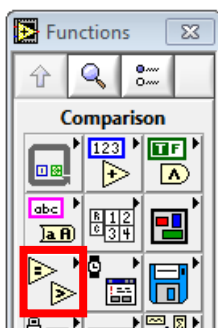


Рисунок 2.2.2.9 – Подсветка в панели функций («*Functions*») раздела элементов сравнения («*Comparison*»)

Для всех случаев, когда требуется сравнение с какими-либо значениями, отличными от нуля, следует выбирать из перечня операции, требующие наличия двух операндов на входе, для случаев сравнения с нулём – операций, требующих на входе один операнд.

Рассматривается простейший случай сравнения: точное соответствие значений элементов друг другу – их эквивалентность («Equal?») (Рисунок 2.2.2.10). Соответствующий элемент располагается в левом верхнем углу окна, содержащего элементы сравнения («Comparison»).

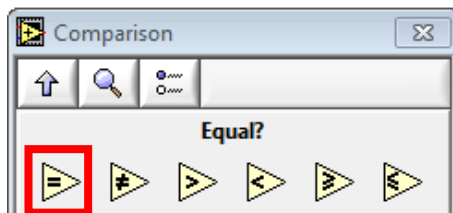


Рисунок 2.2.2.10 – Содержимое раздела функций сравнения («Comparison») с подсветкой оператора однозначного соответствия / проверки на эквивалентность («Equal?»)

Размещаем операцию сравнения на блок-диаграмме ВП (Рисунок 2.2.2.11).

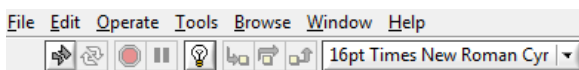


Рисунок 2.2.2.11 – Размещение оператора однозначного соответствия на блок-диаграмме

Слева подключаем к элементу сравнения переменную «Вход» и константу со значением «5», справа – переменную-связку с логическим элементом вывода «Выход».

Не следует и здесь забывать о правилах вежливости программиста: составленный графический код должен обладать читаемостью. К графическому коду предъявляются более строгие требования, определяющие его читаемость. Так, например, при компоновке содержимого блок-диаграммы ВП в *NI LabView* важно стремиться к минимизации изломов линий связи. Избыточные изломы связи отвлекают внимание составителя на себя (подсознание всегда стремится дорабатывать все прямые углы до прямоугольников). Вместе с тем следует стремиться к минимизации области (пространства), занимаемой графическим кодом.

Последнее требование перешло в область составления графического кода из области электроники и схемотехники, где при проектировании печатных плат разработчики стараются минимизировать количество использованного проводника, поскольку каждый сантиметр потраченного материала, как правило, резко повышает стоимость целевой разработки. В учебных заданиях, нацеленных на закрепление навыков работы с *NI LabView*, естественно, создаются только модели вычислительных устройств, однако о минимизации затрат на токопроводящий материал следует привыкать задумываться именно на этапах моделирования (Рисунок 2.2.2.12).

В тот момент, когда у проектировщика вырабатывается привычка минимизации занимаемого пространства элементами схем в процессе моделирования, – это становится серьёзным практическим навыком при разработке реальных электрических схем.

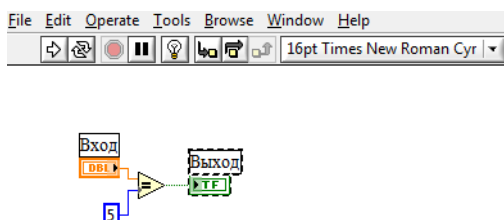


Рисунок 2.2.2.12 – Соединение размещённых на блок-диаграмме ВП элементов между собой

Протестируем созданный ВП. В изображённом на блок-диаграмме ВП алгоритме имеет место разветвление вычислительного процесса. При разветвлении вычислительного процесса в простейшем случае существуют только два исхода (истинный и ложный), потому для доказательства работоспособности нового ВП необходимо подобрать и проверить как минимум два тестовых набора исходных с известными результатами / исходами – продумать схему тестирования, подобрать тестовые примеры. Рекомендуется прочитать / перечитать материал раздела №7 в [3] без выполнения поставленной в нём задачи.

Подадим для начала на вход числового элемента ввода ВП значение равное «2». В этих условиях логическое выражение отвечает ложным результатом – лампочка не загорается (Рисунок 2.2.2.13).

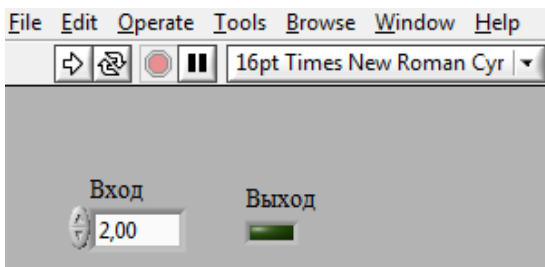


Рисунок 2.2.2.13 – Отклик лампы на значение «2,00», поданное на вход ВП

Затем подадим на числовой элемент ввода ВП значение, равное «5». Здесь логическое выражение отвечает истинным результатом – лампочка загорается (Рисунок 2.2.2.14). Тестирование ВП завершено успешно.



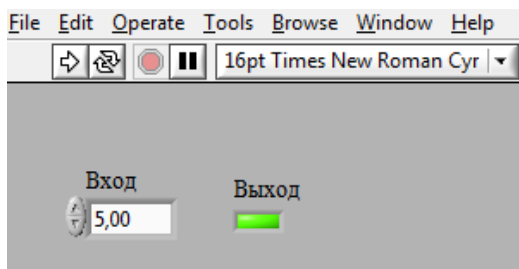


Рисунок 2.2.2.14 – Отклик лампы на значение «5,00», поданное на вход

Рассмотренных выше примеров достаточно для предоставления обучающимся базовых умений и навыков, необходимых для решения задач в сформулированном задании «Разработка ВП "Простой калькулятор"».

## 2.3 Пример выполнения задания

Примеры, собранные в параграфе 2.3 и его разделах, содержат фрагменты ВП, которые обучающиеся могут использовать в качестве основы для разработки своих собственных ВП, заданных по вариантам.

В настоящем Сборнике типовых задач намеренно не рассматриваются готовые решения по созданию ВП согласно сформулированному техническому заданию на разработку. Это сделано с целью исключения шаблонного мышления у обучающихся. Примеры ориентированы на раскрытие познавательной активности обучающихся через действия: «додумать» и/или «развить» начатое. Такая методика обучения базируется на допущении, что самое сложное в любой задаче из любой сферы деятельности – это приступить к её решению.

Благодаря примерам, приведённым в данном Сборнике типовых задач у обучающихся имеется возможность начинать разработку не с чистого листа, а с некоторой предложенной заготовки.

### 2.3.1 Общая часть

В общей части задания «Разработка ВП "Простой калькулятор"» от обучающихся требуется рассмотреть функционирование четырёх арифметических операций, каждая из которых выполняется над парой операндов. Результат выполнения каждой операции должен выводиться в свой отдельный элемент вывода.

На Рисунках 2.3.1.1 и 2.3.1.2 показан пример ВП, реализующего только одну арифметическую операцию – сложение двух операндов. Передняя панель ВП содержит отработанный тестовый пример. Действительно,  $5 + 3 = 8$  – тестирование пройдено успешно.

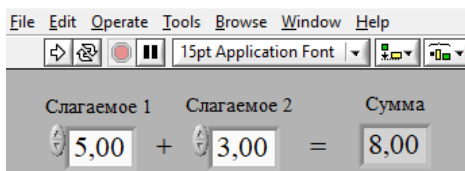


Рисунок 2.3.1.1 – Тестовый пример, рассчитанный при отладке фрагмента ВП, составляемого в общей части задания

На блок-диаграмме ВП (Рисунок 2.3.1.2) показано, что операнды «Слагаемое 1» и «Слагаемое 2», поданные на вход операции «Сложение», формируют на её выходе значение, равное «8», которое подаётся на вход числового элемента вывода. Числовой элемент вывода отображает полученное значение на передней панели ВП.

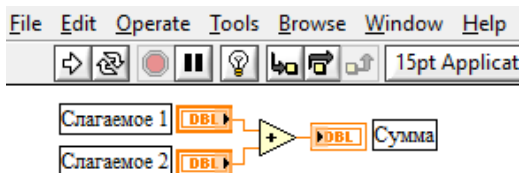


Рисунок 2.3.1.2 – Реализация суммирования двух операндов на блок-диаграмме ВП

### 2.3.2 Индивидуальная часть (переменные)

Индивидуальная часть задания «Разработка ВП "Простой калькулятор"» разбита на две составляющие, одна из которых посвящена созданию ВП, показывающего параметрический расчёт арифметического выражения, состоящего из нескольких операндов и операций, а также констант (в отдельных вариантах задания).

Для примера в качестве арифметического выражения взято: « $5 * (a / b) ^ 2$ ». Оно введено на передней панели фрагмента ВП в качестве строки текста (Рисунок 2.3.2.1). Вместе с ним на переднюю панель вынесены элемент вывода результата, а также операнды « $a$ » и « $b$ », входящие в состав арифметического выражения в качестве параметров, значения которых должны вводиться пользователем.

На передней панели ВП отработан тестовый пример. Действительно,  $5 * (9 / 3) ^ 2 = 5 * 3 ^ 2 = 5 * 9 = 45$ .

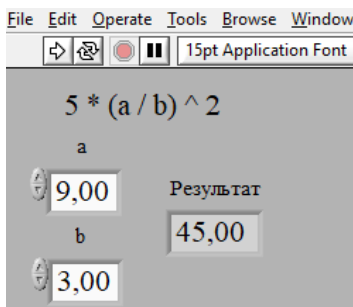


Рисунок 2.3.2.1 – Тестовый пример, рассчитанный при отладке фрагмента ВП, составляемого в индивидуальной части задания (переменные)

Декомпозиция заданного арифметического выражения позволяет сформулировать алгоритм решения задачи:

1. Исходными данными является константа со значением «5», параметрами являются переменные с именами « $a$ » и « $b$ ».

2. Первым всегда выполняется действие, заключённое в скобки, потому деление « $a$ » на « $b$ » должно быть выполнено в первую очередь.

3. Значение, полученное в скобках, должно быть возведено в квадрат. Для того, чтобы не усложнять программу специальными функциями, для возведения в квадрат следует использовать эквивалентную операцию перемножения полученного значения самого на себя.

4. Полученный результат должен быть умножен на константу со значением «5».

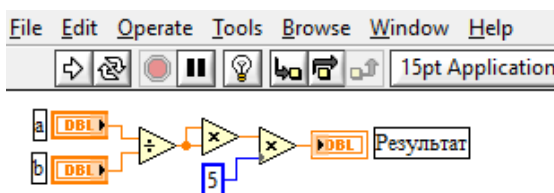


Рисунок 2.3.2.2 – Реализация на блок-диаграмме ВП операций, проводимых над константой и парой переменных с выводом результата расчёта на переднюю панель ВП

Обучающимся следует обратить внимание на то, что в данной программной реализации не отлавливается ошибка, связанная с делением на ноль и близкие к нулю значения. Для закрепления изученного материала предлагается доработать ВП в этом направлении самостоятельно.

### 2.3.3 Индивидуальная часть (константы)

Вторая составляющая индивидуальной части задания «Разработка ВП "Простой калькулятор"» посвящена созданию ВП, в котором константы увязаны между собой операциями. Как правило, это числовые константы, однако в отдельных случаях обучающимся может потребоваться использование специальных констант, таких как число «Пи» и/или «экспонента» (Рисунок 2.3.3.1).

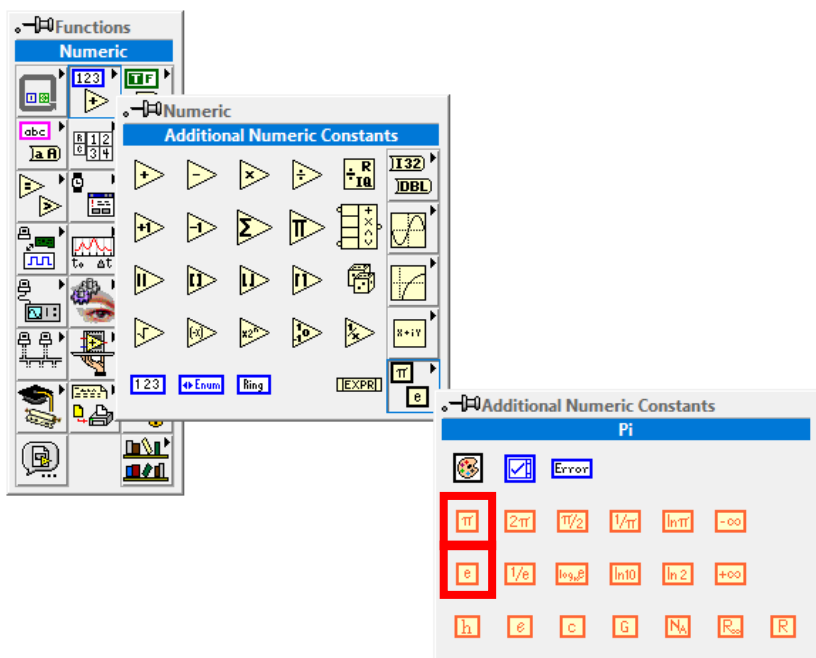


Рисунок 2.3.3.1 – Маркеры размещения специальных констант

Вместе с тем в большинстве сложных арифметических выражений, заданных по вариантам, довольно частой является операция взятия корня степени « $N$ ». Эту операцию можно найти в перечне логарифмических функций («*Logarithmic*») (Рисунок 2.3.3.2)

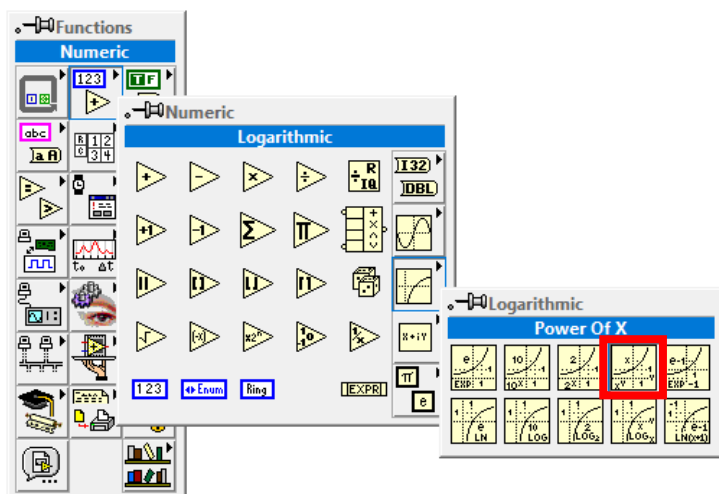


Рисунок 2.3.3.2 – Маркеры размещения функции возведения значения, поданного на вход «X» в степень, поданную на вход «Y»

На Рисунке 2.3.3.3 представлен вид фрагмента сложного арифметического выражения, размещённого на передней панели ВП. Для удобства восприятия пользователем изображения-вставки фон передней панели ВП изменён с серого на белый. Отображён результат расчёта того же тестового примера, который рассмотрен в задании «Приобретение навыков работы с калькулятором *Microsoft Windows*» из [3]. Результат намеренно выведен с той же точностью до трёх знаков после десятичного разделителя.

The image shows a software interface with a menu bar (File, Edit, Operate, Tools, Browse, Window) and a toolbar. Below the toolbar, the expression  $5 \sqrt{\left(\frac{1}{0.13}\right)^3}$  is displayed, followed by an equals sign and the result '3,401'. The text 'Результат' is above the result.

Рисунок 2.3.3.3 – Тестовый пример, рассчитанный при отладке фрагмента ВП. Индивидуальная часть (константы)

По итогам анализа фрагмента сложного арифметического выражения можно выполнить упрощение, рассматривая частное  $(1 / 0,13)$  в степени  $(3 / 5)$ . Это позволяет отказаться от двух операций умножения (или дополнительной операции возведения в степень). На Рисунке 2.3.3.4 представлена блок-диаграмма такого ВП с уже выполненным упрощением.

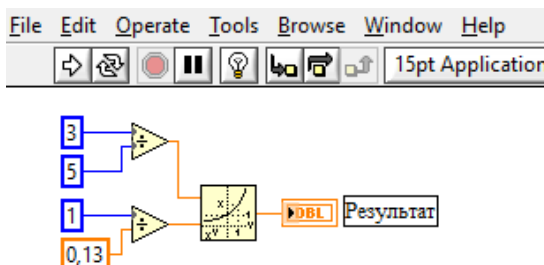


Рисунок 2.3.3.4 – Реализация на блок-диаграмме ВП операций, проводимых над константами с выводом результата расчёта на переднюю панель ВП

## 2.4 Варианты индивидуального задания

### 2.4.1 Переменные в ВП

В дополнительном файле \*.vi (придумать имя согласно Приложению 3) создать развитие структуры «Простого Калькулятора» посредством добавления в неё элементов ввода и вывода информации, а также индикации (в случае необходимости), достаточных для демонстрации результата расчёта выражения, заданного для варианта из Таблицы 2.4.1.1. Варианты распределяются в соответствии с номерами обучающихся в списке группы.

**Без особой необходимости графический код на блок-диаграмме не усложнять!** Например, операция возведения в целую степень не должна вызывать потребности у автора виртуального прибора в использовании специальной функции возведения числа в некоторую степень «N».

Таблица 2.4.1.1 – Варианты индивидуального задания для закрепления навыков работы с переменными в ВП

Вар.	Выражение	Логический элемент вывода
01	$(a + b + c + d) * (-a - b - c - d)$	Не требуется
02	$(a + b + c)^2$	Не требуется
03	$(a - b + c - d) * (-a + b - c - d)$	Не требуется
04	$(b / a) - (d / c)$	Требуется
05	$(a + (b - d) * c)^2$	Не требуется
06	$(d / (b + a)) - (c / (d + a))$	Требуется
07	$3 * a + b^2 - c + 5 * d^3$	Не требуется
08	$-((b + c) / (a + d))$	Требуется
09	$(a + (b * c) / d) / (a / (b * d) - c)$	Требуется
10	$((b + 2) * (a + 4) / (d + 6) * (c + 8))$	Требуется
11	$-((b + c) * (a + d) / c^2)$	Требуется
12	$10 * a + 15 * b + 20 * c + 25 * d$	Не требуется
13	$7 * (a + b + c + d) * (-a - 7 * b - c - d)$	Не требуется
14	$(a + b + c + d) / (-a - b - c - d)$	Требуется
15	$(a + b + c + d)^3$	Не требуется
16	$(a - b + c - d) / (-a + b - c - d)$	Требуется
17	$(a / b) + (c / d)$	Требуется
18	$(a * (b - d) + c)^2$	Не требуется
19	$(a / (c + a)) + (b / (d * a))$	Требуется
20	$((b / a) / (d / c)) - ((a / b) / (c / d))$	Требуется
21	$5 * a + b^2 + c - 3 * d^3$	Не требуется
22	$-((a + b) / (d - c))$	Требуется
23	$(a - (b * c) / d) * (a / (b * d) + c)$	Требуется
24	$10 * d - 5 * c + 20 * a - 15 * b$	Не требуется
25	$-((b - c) * (a + d) / (c + 4))$	Требуется
26	$3 * (a + c + d) * (-7 * b - d)$	Не требуется
27	$(a + b + c + d)^4$	Не требуется
28	$(a * (c + a)) / (b * (d - a))$	Требуется
29	$(a / b)^2 + (c / d)^2$	Требуется
30	$(a^2 + b^2 + c^2)$	Не требуется



<b>31</b>	$((a + b - d) * c)^2$	Не требуется
<b>32</b>	$((a + b * c) / d) / ((a / b) * d) - c$	Требуется
<b>33</b>	$(a + b + c + d) / (-a - (b - c) - d)$	Требуется
<b>34</b>	$(a + (b + c)^3 + d)$	Не требуется
<b>35</b>	$(a - (b + c)^2)$	Не требуется

### 2.4.2 Константы в ВП

В ещё одном дополнительном файле \*.vi (придумать имя согласно Приложению 3) создать независимый от двух предыдущих ВП иной, который смоделировал бы расчёт правильного ответа на сложное арифметическое выражение, заданное по варианту.

Графический пользовательский интерфейс должен содержать в своём составе в удобной для чтения форме исходное сложное арифметическое выражение, заданное по варианту, а также элемент вывода, выдающий правильный ответ на заданное сложное арифметическое выражение.

$$1 \text{ Вар. } \sqrt[3]{\frac{5 \sqrt{3.475 + \left(5 \frac{1}{11}\right)} - \sqrt{\frac{\left(2 + \frac{124.33}{70.3}\right)}{9.1}} + \frac{\left(2 + \frac{1}{4} \frac{1}{0.75}\right)}{9 - \frac{50}{2 + 2.2 \cdot \frac{10}{22}}} - \frac{1}{\sqrt{\pi}}}$$

$$2 \text{ Вар. } \frac{26}{\sqrt[3]{\frac{(99.06 - 40.81) \cdot 2}{3.14} - \sqrt{\frac{\frac{3.0}{0.2} + \frac{4.2}{0.1}}{\left(\frac{1}{0.3} - 1.2\right)}}}} + \frac{\frac{12}{21}}{\frac{8}{28}} + \left(\frac{1 + \sqrt[3]{10}}{33.15}\right)$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$3 \text{ Вар. } \frac{\left(\frac{4.2 + 11 \frac{5}{7}}{\frac{2}{6} - \frac{1}{7}}\right) \cdot 1.78}{\frac{32}{41}} - \sqrt{\frac{\left(4.5 - 1 \frac{7}{9}\right) \cdot 12}{2 \frac{3}{20} - 0.45}} + \left[\frac{1}{\frac{1 + \frac{\pi}{5\sqrt{3}}}{1 + \frac{e}{\sqrt{2}}}}\right]^{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$4 \text{ Вар. } \left[\sqrt{\frac{\frac{3 - 2 \frac{1}{5}}{0.002}}{\sqrt{\left(3 \frac{1}{4} + 0.62\right)} \cdot 4}} - \sqrt[3]{\frac{\left(0.3 - \frac{3}{20}\right) \cdot 1 \frac{1}{5}}{\left(2.25 + 2 \frac{3}{5}\right) \cdot \frac{1}{8}}} \cdot \left(\frac{1}{5}\right)^{\sqrt{5}}\right]$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$5 \text{ Вар. } \sqrt[5]{\frac{\sqrt[4]{\frac{\left(1.1 + 1 \frac{5}{7}\right) \cdot 3.0127}{\frac{2}{3} - \frac{1}{11}}} - \sqrt[3]{\frac{\left(4.25 - 1 \frac{5}{11}\right) \cdot 11}{3 \frac{3}{12} - 0.45}}}{\frac{9}{11}}} + \left(\frac{1}{\frac{\pi}{\sqrt[3]{8}}}\right)^{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

Рисунок 2.4.2.1 – Варианты 1-5 индивидуального задания для закрепления навыков работы с константами в ВП

6 Вар.

$$-\sqrt[3]{4.127} - \sqrt{\frac{\left(\frac{1.95 + \frac{54.23}{24.5}}{0.95}\right)}{\frac{\sqrt[3]{2\frac{3}{7} - 1.0567}}{4.315}}} + \frac{\left(2 + \frac{1}{2} \cdot \frac{1}{0.25}\right)}{3 - \frac{33}{1 + 3.3 \cdot \pi^2}} + \frac{1}{\sqrt{\pi}}$$

7 Вар.

$$\sqrt{\frac{34\frac{15}{24} - \sqrt[5]{32\frac{9}{13}}}{0.182 \cdot \sqrt{0.252}} + 3\frac{1}{5} \cdot 3\frac{2}{3}} - \sqrt[7]{\left(\frac{1}{0.013}\right)^3} \cdot 2\frac{7}{121}$$

$$\sqrt[3]{\left(3\frac{3}{4} \cdot 3\frac{3}{5} \cdot 3\frac{3}{6}\right)}$$

8 Вар.

$$\sqrt{\frac{\sqrt[5]{\left(100\frac{5}{15} - 96\frac{3}{7}\right)}}{11.7} + 0.1234}{\left(1\frac{3}{4} \cdot 1\frac{5}{6}\right)} + \sqrt[5]{\left(\frac{2}{2.96}\right)^3} + \frac{e}{3}$$

9 Вар.

$$\sqrt[5]{\frac{(7.2 - 1.7) \cdot \sqrt[2]{4\frac{6}{13}}}{\frac{\left(5\frac{2}{5} - 1\frac{1}{8}\right)}{\frac{13}{17}}}} - \sqrt{e} + \sqrt[3]{\left(2\frac{5}{8}\right) - \frac{\left(4 + \frac{4}{4\sqrt[4]{16 - 9\frac{15}{18}}}\right) \cdot \frac{1}{4}}{2\frac{1}{5}}} + \sqrt[4]{4.255}$$

10 Вар.

$$\sqrt[4]{\frac{(2.5 - 0.8) \cdot \sqrt[3]{2\frac{1}{3}}}{\frac{(2.5 - 1\frac{14}{15})}{\frac{3}{7}}}}} + \sqrt[3]{8 - \frac{\left(95 + \frac{7}{5\sqrt[5]{30 - 93\frac{5}{18}}}\right) \cdot 2\frac{1}{4}}{2\frac{1}{5}}} + \sqrt[6]{\pi}$$

11 Вар.

$$\sqrt{\frac{\left[\left(152\frac{3}{4}\right) - \left(148\frac{3}{8}\right)\right] \cdot 0.3}{0.2}} + \frac{\left[\left(172\frac{5}{6}\right) - \left(170\frac{1}{3}\right) + \left(3\frac{5}{12}\right)\right]}{0.8 \cdot 0.25} - \sqrt[5]{6}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

Рисунок 2.4.2.2 – Варианты 6-11 индивидуального задания для закрепления навыков работы с константами в ВП

$$12 \text{ Вар. } \sqrt[3]{\frac{\sqrt[5]{\frac{85\frac{7}{30}-83\frac{5}{18}}{2\frac{1}{3}}}}{\frac{0.04}{0.02+\frac{1}{5}}} + \frac{\left(140\frac{7}{30}-138\frac{5}{12}\right)}{18\frac{1}{6}}} \cdot \frac{1}{3} - \sqrt[3]{\frac{1}{\pi}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$13 \text{ Вар. } \frac{\sqrt{\frac{\left(58\frac{4}{15}-56\frac{7}{24}\right)}{0.8}} + 2\frac{1}{9} \cdot 0.225}{8\frac{3}{4} \cdot \frac{3}{5}} - \sqrt[5]{\left(\frac{1}{0.13}\right)^3}$$

$$14 \text{ Вар. } \frac{\sqrt{(2.7-0.8) \cdot 2\frac{1}{3}}}{\frac{(5.2-1.4)}{\sqrt[5]{\frac{3}{7}}}} - 2.625 + 8\frac{9}{11} - \sqrt[3]{\frac{\left(1.6+\frac{154.66}{70.3}\right)}{\frac{1.9}{\frac{2\frac{2}{5}-1.3}{4.3}}}} + 5^{0.12}$$

$$15 \text{ Вар. } \frac{26}{\sqrt[7]{\frac{\frac{3}{0.2-0.1}}{2.5 \cdot (0.8+1.2)} + \frac{(34.06-33.81) \cdot 4}{6.84}}}} + \sqrt[5]{\frac{\frac{\pi}{3}}{\frac{4}{21}} + \left[\frac{(1+5\sqrt{3})}{3.5}\right]^{\frac{1}{3}}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$16 \text{ Вар. } \left[ \frac{\frac{(6-4\frac{1}{2})}{0.03}}{\sqrt{\left(\frac{3\frac{1}{20}-2.65}{1.88+2\frac{3}{25}}\right) \cdot 4 + \frac{2}{5}}} - \frac{4\sqrt{\left(0.3-\frac{3}{20}\right) \cdot 1\frac{1}{2}}}{\left(\frac{1}{5}\right)^{\sqrt{3}}} \right]$$

$$17 \text{ Вар. } \left[ \frac{\sqrt[5]{\frac{(2.7-0.8) \cdot \sqrt[3]{2\frac{1}{3}}}{\frac{(5.2-1.4)}{\frac{3}{7}}}} + 8\frac{9}{11}}{\frac{95+\frac{7}{30-93\frac{5}{18}}}{2^{1.2}}} \cdot 2\frac{1}{4} + \frac{2 \cdot \pi}{3} \right]$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

Рисунок 2.4.2.3 – Варианты 12-17 индивидуального задания для закрепления навыков работы с константами в ВП

$$18 \text{ Вар. } \left[ \sqrt[2]{\left( \frac{85 \frac{7}{30} - 83 \frac{5}{18}}{3 \sqrt[2]{\frac{2}{3}}} \right) + \frac{\left( \frac{140 \frac{7}{30} - 138 \frac{5}{12}}{18 \frac{1}{6}} \right)}{0.02 + \frac{1}{5}}} \right] \cdot \frac{1}{3} - \sqrt[3]{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$19 \text{ Вар. } \left[ \sqrt[4]{\left( \frac{85 \frac{7}{30} - 83 \frac{5}{18}}{2 \frac{2}{3}} \right) + \frac{\left( \frac{140 \frac{7}{30} - 138 \frac{5}{12}}{18 \frac{1}{6}} \right)}{0.02 + \frac{1}{5}}} \right] \cdot \frac{1}{45} - \sqrt[5]{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$20 \text{ Вар. } \left[ \frac{\left( \frac{85 \frac{7}{30} - 83 \frac{5}{18}}{2 \frac{2}{3}} \right)}{0.04} + \sqrt[2]{\frac{\left( \frac{14 \frac{4}{30} - 10 \frac{5}{12}}{18 \frac{1}{6}} \right)}{0.02 + \frac{1}{5}}} \right] \cdot \frac{1}{3} - \sqrt[3]{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$21 \text{ Вар. } 8 \frac{9}{11} - \frac{1}{\sqrt{\pi}} - 2.625 - \sqrt[2]{\frac{\left( \frac{1.6 + \frac{154.66}{70.3}}{1.9} \right)}{\left( \frac{2 \frac{2}{3} - 1.3}{4.3} \right)}} + \frac{\left( 1 + \frac{1}{2} \cdot \frac{1}{0.25} \right)}{6 - \frac{46}{1 + 2.2 \cdot 10}}$$

$$22 \text{ Вар. } \frac{(2.7 - 0.8) \cdot 2 \frac{1}{3}}{\frac{3}{7}} - 2.625 + 8 \frac{9}{11} - \frac{\left( \frac{7}{30 - 93 \frac{5}{18}} \right) \cdot 2 \frac{1}{4}}{0.2}$$

$$23 \text{ Вар. } \sqrt[5]{\frac{(2.7 - 0.8) \cdot 2 \frac{1}{3}}{\frac{3}{7}}} - \sqrt[2]{2.625 + 8 \frac{9}{11}} - \frac{\left( 95 + \frac{7}{30 - 93 \frac{5}{18}} \right) \cdot 2 \frac{1}{4}}{2^{1.2}} + \sqrt[6]{\pi}$$

Рисунок 2.4.2.4 – Варианты 18-23 индивидуального задания для закрепления навыков работы с константами в ВП

24 Вар.

$$\frac{26}{\frac{(34.06 - 33.81) \cdot 4}{6.84} - \sqrt{\frac{3 + \frac{4.2}{0.1}}{\frac{\left(\frac{1}{0.3} - 2.7\right)}{4.12}}}} + \frac{\frac{2}{3}}{\frac{4}{21}} + \left(\frac{1 + \sqrt[3]{5}}{3.5}\right)^{\frac{1}{4}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

25 Вар.

$$\sqrt[3]{\frac{\sqrt{\frac{\left(\frac{40 \cdot \frac{7}{30}\right) - \left(38 \cdot \frac{5}{12}\right)}{1.9} + \left(\frac{7}{8} - \frac{7}{30}\right) \cdot \left(1 \cdot \frac{9}{11}\right)} \cdot 5.143}{0.008}} - \frac{1}{\sqrt{2.24^4}}$$

26 Вар.

$$\sqrt[5]{\frac{\frac{85 \cdot \frac{7}{30} - 83 \cdot \frac{5}{18}}{2 \cdot \frac{2}{3}} + \frac{\left(140 \cdot \frac{7}{30} - 138 \cdot \frac{5}{12}\right)}{18 \cdot \frac{1}{6}}}{0.04 + \frac{1}{5}}} \cdot \frac{1}{3} - \sqrt[3]{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

27 Вар.

$$\sqrt[5]{\frac{\frac{85 \cdot \frac{7}{30} - 83 \cdot \frac{5}{18}}{2 \cdot \frac{2}{3}} + \frac{\left(140 \cdot \frac{7}{30} - 138 \cdot \frac{5}{12}\right)}{18 \cdot \frac{1}{6}}}{0.04 + \frac{1}{5}}} \cdot \frac{1}{3} - \sqrt[3]{\frac{1}{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

28 Вар.

$$\frac{26}{\frac{(34.06 - 33.81) \cdot 4}{6.84} - \sqrt{\frac{3 + \frac{4.2}{0.1}}{\frac{\left(\frac{1}{0.3} - 2.7\right)}{4.12}}}} + \frac{\frac{2}{3}}{\frac{4}{21}} + \left(\frac{1 + \sqrt[3]{5}}{3.5}\right)^{\frac{1}{5}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

Рисунок 2.4.2.5 – Варианты 24-28 индивидуального задания для закрепления навыков работы с константами в ВП

$$29 \text{ Вар.} \quad \frac{\sqrt{(2.7 - 0.8) \cdot 2 \frac{1}{3}}}{\sqrt[5]{\frac{3}{7}}} - \frac{\sqrt[3]{\frac{1.6 + \frac{154.66}{70.3}}{1.9}}}{\frac{(\frac{2.2}{5} - 1.3)}{4.3}} + 5^{0.0123}$$

$$30 \text{ Вар.} \quad \left[ \frac{(\frac{6 - 4 \frac{1}{2}}{0.03})}{\sqrt{\frac{3 \frac{1}{2} - 2 \cdot 15 + \frac{2}{5}}}} - \frac{\sqrt[4]{0.3 - \frac{3}{12} \cdot \frac{1}{2}}}{\left(18.8 + 7 \frac{3}{25}\right) \cdot \frac{1}{8}} \right] \cdot \left(\frac{1}{5}\right)^{\sqrt{3}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$31 \text{ Вар.} \quad \sqrt{\frac{\left(52 \frac{3}{4} - 48 \frac{3}{8}\right)}{0.7}} + \frac{\left(72 \frac{5}{6} - 37 \frac{1}{3} + 13 \frac{5}{12}\right)}{0.8 - 0.225} - 8\sqrt{2}$$

$$32 \text{ Вар.} \quad \frac{1}{\frac{(5.2 - 1.4)}{\frac{3}{7}}} + e - \frac{\left(9 + \frac{7}{1 - 20 \frac{5}{18}}\right) \cdot 3 \frac{3}{4}}{0.2} + \pi$$

$$33 \text{ Вар.} \quad \frac{1}{\left[ \frac{(34.06 - 33.81) \cdot 4}{\frac{6.84}{28.57 - 25.15}} - \frac{\sqrt[3]{3 + \frac{4.2}{0.1}}}{\frac{\frac{1}{0.3} - 2.7}{1^{2.25}}} \right]} + \frac{\frac{4}{21}}{\frac{2}{3}} + \left( \frac{1 + \sqrt[3]{5}}{3.5} \right)^{\frac{1}{5}}$$

[ ] - те же скобки, что и ( ). Проставлены для удобства чтения выражения.

$$34 \text{ Вар.} \quad \left( \sqrt[4]{\frac{\frac{7}{18} - \frac{5}{30}}{\frac{2}{3}} + \frac{\frac{7}{30} - \frac{5}{12}}{\frac{1}{6}}} \right) \cdot \frac{1}{45} - \sqrt[5]{\frac{1}{3}}$$

$$35 \text{ Вар.} \quad \frac{26}{\frac{(34.06 - 33.81)}{\sqrt[5]{28.57 - 25.15}}} + \frac{\frac{2}{3}}{\frac{4}{21}} + \left( \frac{1 + \sqrt[3]{5}}{3.5} \right)^{\frac{1}{4}}$$

Рисунок 2.4.2.6 – Варианты 29-35 индивидуального задания для закрепления навыков работы с константами в ВП

## 2.5 Пример компоновки графического пользовательского интерфейса ВП

### 2.5.1 Общая часть

На Рисунке 2.5.1.1 представлен пример реализации передней панели ВП, который необходимо разработать в общей части задания. Наименования операндов и результатов могут быть изменены, как изменено может быть и расположение элементов управления и надписей относительно друг друга.

Интерфейс может быть дополнен элементами управления, в частности, элементом ввода интересующей точности производимых вычислений.

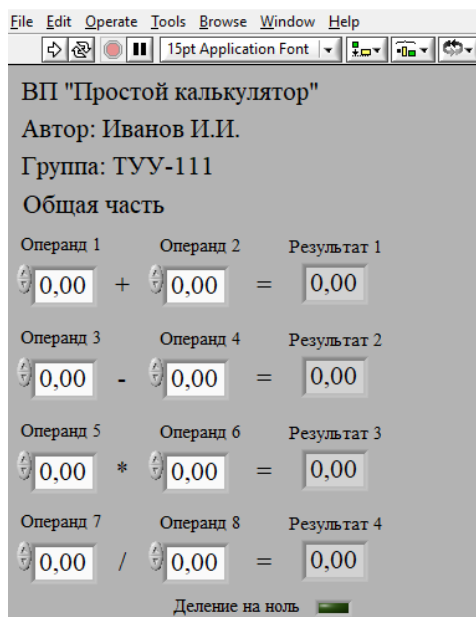


Рисунок 2.5.1.1 – Опорный пример передней панели ВП, создаваемого в общей части задания



## 2.5.2 Переменные в ВП

На Рисунке 2.5.2.1 представлен пример реализации передней панели ВП, который необходимо разработать в одной из составляющих индивидуальной части задания.

При реализации ВП избыточность может быть устранена из приведённого примера, в котором одно и то же арифметическое выражение представлено в двух различных вариантах записи.

Следует так же напомнить, что не все варианты требуют контроля нулевого (или близкого к нулю) знаменателя. В некоторых вариантах, напротив, имеет место контроль нуля (или близкого к нулю значения) для нескольких знаменателей в составе заданного арифметического выражения.

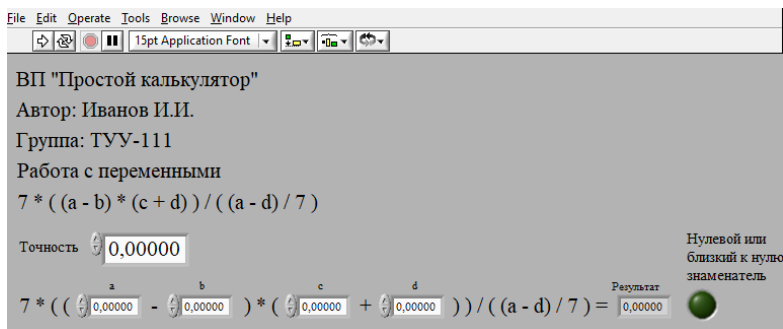


Рисунок 2.5.2.1 – Опорный пример передней панели ВП, создаваемого в индивидуальной части задания, связанной с закреплением навыков работы с переменными

### 2.5.3 Константы в ВП

На Рисунке 2.5.3.1 представлен пример реализации передней панели ВП, который необходимо разработать во второй из составляющих индивидуальной части задания.

При реализации ВП не обязательно указывать номер варианта обучающегося в списке группы, поскольку эта информация содержится на титульном листе отчёта о выполненной работе.

В остальном, передняя панель ВП должна получиться близкой к представленному изображению (см. Рисунок 2.5.3.1). Точность вычислений можно выставить выше, ниже или равной представленной в зависимости от особенностей получаемого результата.

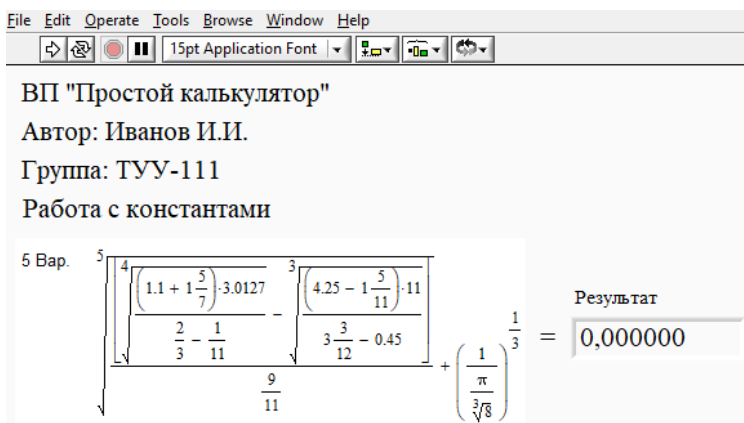


Рисунок 2.5.3.1 – Опорный пример передней панели ВП, создаваемого в индивидуальной части задания, связанной с закреплением навыков работы с константами

## 2.6 Рекомендации по размещению информации на блок-диаграмме ВП

### 2.6.1 Общая часть

На Рисунке 2.6.1.1 представлено начальное приближение к составлению графического кода. В общем случае расположение и количество операторов может быть изменено, количество элементов во входной и выходной информации может быть дополнено.

Так, например, данный набор элементов не учитывает точности производимых вычислений, хотя согласно алгоритму, представленному в разделе 2.7.1, и формулировке из раздела 2, точность требуется учитывать.

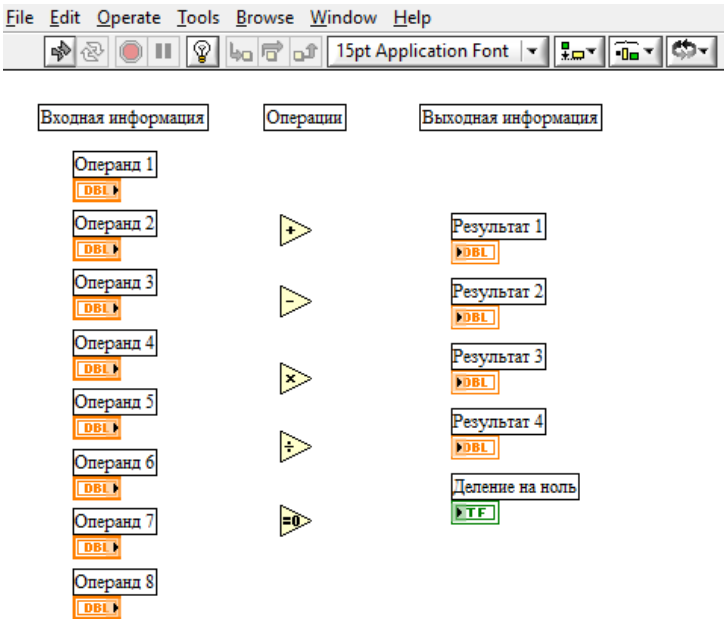


Рисунок 2.6.1.1 – Рекомендация по начальной компоновке элементов на блок-диаграмме ВП, создаваемого в общей части задания

## 2.6.2 Переменные в ВП

На Рисунке 2.6.2.1 представлено начальное приближение к составлению графического кода, основанного на некотором примере арифметического выражения (оно то же, что представлено в разделе 2.5.2).

В общем случае перечень констант и количество операндов в составе входной информации будет меняться в зависимости от варианта обучающегося в списке группы. Так же будет меняться состав и количество операторов, представленных в центральной части блок-диаграммы ВП, но все они в любом случае будут выбираться из множества арифметических: «сложить», «вычесть», «умножить», «разделить».

В отдельных вариантах потребуется либо полностью убрать, либо увеличить количество элементов вывода, предназначенных для визуального контроля получения нуля или близких к нулю значений в знаменателе (знаменателях).

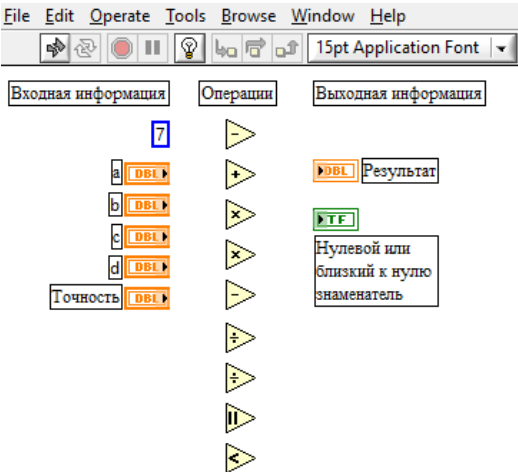


Рисунок 2.6.2.1 – Рекомендация по начальной компоновке элементов на блок-диаграмме ВП, создаваемого в индивидуальной части задания, связанной с переменными

### 2.6.3 Константы в ВП

На Рисунке 2.6.3.1 представлено начальное приближение к составлению графического кода, основанного на примере арифметического выражения для 5-го варианта из перечня, представленного в разделе 2.4.2.

Здесь схема, наборы данных и операций будут индивидуальными для каждого варианта.

В примере предпринята попытка исключить повторные упоминания значений в перечне констант, а повторное использование применять на уровне разветвления линий связи.

В общем случае константы могут дублироваться и, вероятно, этот подход может оказаться выгоднее с точки зрения экономии применяемого проводника. Каждый разработчик должен принять для себя решение, которая из схем реализации ему ближе и понятнее.

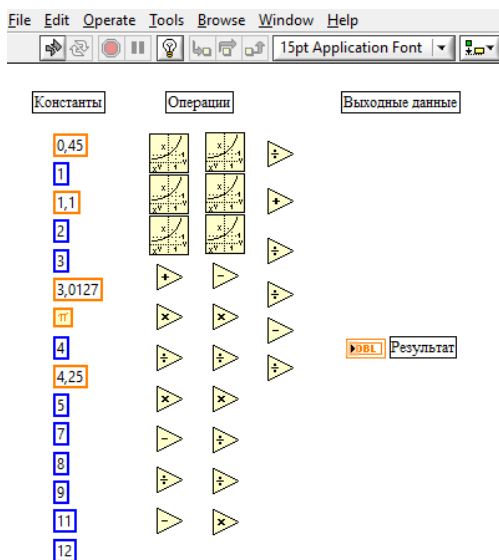


Рисунок 2.6.3.1 – Рекомендация по начальной компоновке элементов на блок-диаграмме ВП, создаваемого в индивидуальной части задания, связанной с константами

## 2.7 Блок-схемы алгоритмов для типовых решений

### 2.7.1 Общая часть

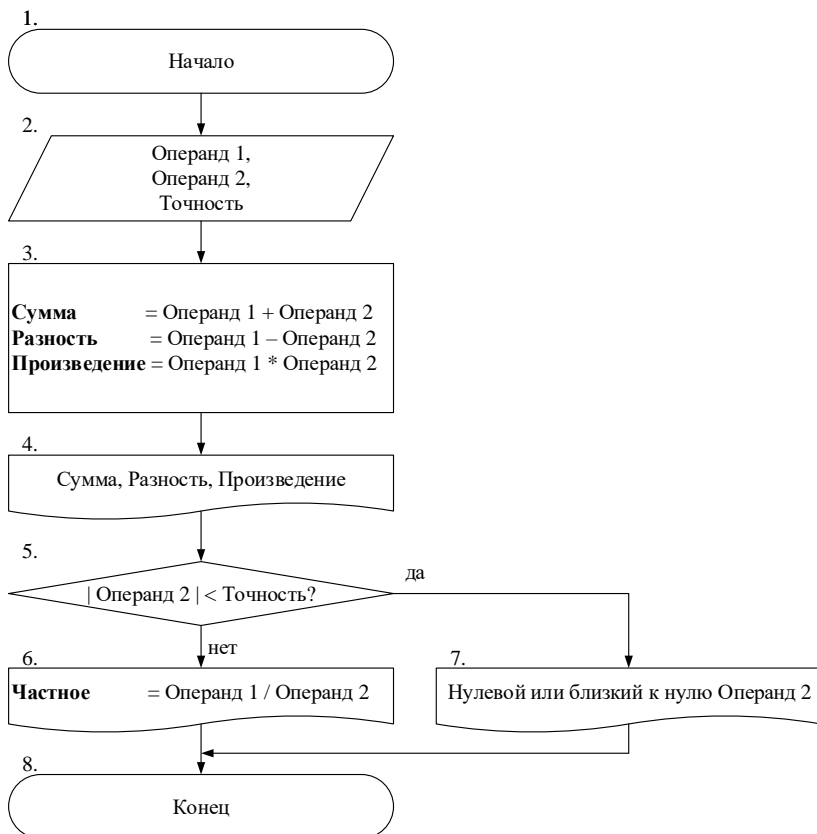


Рисунок 2.7.1.1 – Схема алгоритма для ВП из общей части

Описание алгоритма, представленного на Рисунке 2.7.1.1:

В блоке 1 – обозначено начало вычислительного процесса, обрабатываемого ВП для поиска решений задачи, сформулированной в общей части задания.

В блоке 2 – объявлен перечень исходной информации, состоящей из пары операндов и величины учитываемой / заданной точности производимых вычислений.

В блоке 3 – вычисляется сумма, разность и произведение для пары операндов.

В блоке 4 – результаты произведённых в блоке 3 расчётов выводятся на переднюю панель ВП.

В блоке 5 – перед тем, как перейти к расчёту частного для пары операндов, необходимо проверить значение операнда, стоящего в знаменателе. В том случае, если это значение равно нулю (или располагается в ближайших окрестностях нуля ниже порога учитываемой / заданной точности производимых вычислений), то переходить к блоку 7, в ином случае – переходить к блоку 6.

В блоке 6 – вычисляется частное для пары операндов и результат вычисления выводится на переднюю панель ВП.

В блоке 7 – частное не рассчитывается. Вместо этого на логический элемент вывода, сообщающий о нулевом (или близком к нулю) знаменателе поступает сигнал об ошибке – лампочка загорается.

В блоке 8 – обозначено завершение вычислительного процесса, отрабатываемого ВП для поиска решений задачи, сформулированной в общей части задания.

## 2.7.2 Переменные в ВП

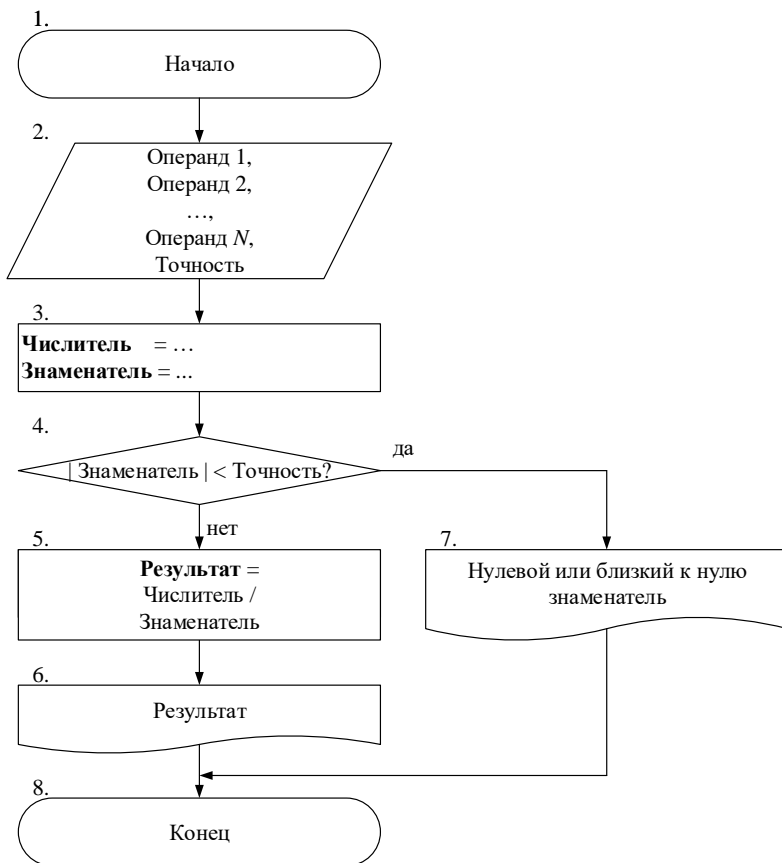


Рисунок 2.7.2.1 – Схема алгоритма для ВП из индивидуальной части (переменные)

Описание алгоритма, представленного на Рисунке 2.7.2.1:

В блоке 1 – обозначено начало вычислительного процесса, обрабатываемого ВП для поиска решения задачи, сформулированной в индивидуальной части задания для отработки навыков использования переменных.



В блоке 2 – объявлен перечень исходной информации, состоящей из требуемого согласно заданию количества операндов и величины учитываемой / заданной точности производимых вычислений.

В блоке 3 – вычисляются отдельно числитель и знаменатель арифметического выражения (рассмотрена ситуация, когда ни числитель, ни знаменатель не содержит в своём составе операцию деления).

В блоке 4 – в том случае, если знаменатель арифметического выражения получился равным нулю или незначимым с точки зрения указанной величины учитываемой / заданной точности, то перейти к блоку 7, в ином случае – к блоку 5.

В блоке 5 – выполняется вычисление результата при делении числителя на знаменатель.

В блоке 6 – выполняется вывод результата деления на переднюю панель ВП.

В блоке 7 – результат деления не вычисляется. Вместо этого на логический элемент вывода ВП подаётся сигнал об ошибке. Лампочка нуля или близкого к нулю значения в знаменателе загорается.

В блоке 8 – обозначено завершение вычислительного процесса, обрабатываемого ВП для поиска решения задачи, сформулированной в индивидуальной части задания для отработки навыков использования переменных.

Рассмотрен общий случай, когда заданное арифметическое выражение содержит как числитель, так и знаменатель. В частных случаях, когда в арифметических выражениях отсутствуют операции деления, блок-схемы алгоритмов будут получаться более короткими и в них не потребуется учёт / ввод значения точности производимых вычислений.

### 2.7.3 Константы в ВП

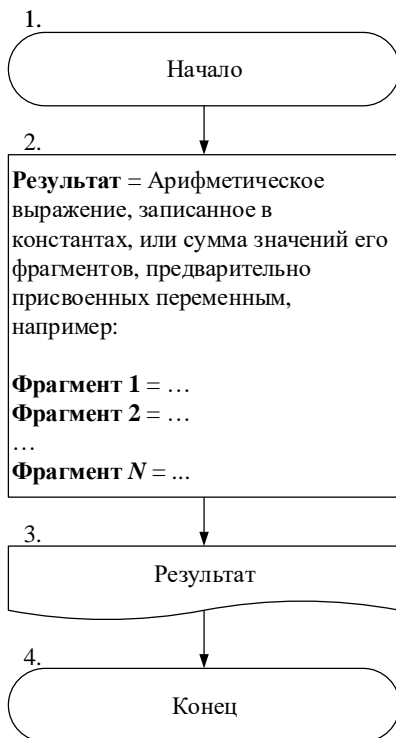


Рисунок 2.7.3.1 – Схема алгоритма для ВП из индивидуальной части (константы)

Описание алгоритма, представленного на Рисунке 2.7.3.1:

В блоке 1 – обозначено начало вычислительного процесса, обрабатываемого ВП для поиска решения задачи, сформулированной в индивидуальной части задания для отработки навыков использования констант.

В блоке 2 – выполняется расчёт сложного арифметического выражения по действиям. В рамках вычислительного процесса ВП разбиение арифметического выражения на фрагменты,

результаты вычисления которых выводятся в дополнительные числовые элементы вывода, не требуется, но обучающиеся могут реализовать это и оставить для нужд тестирования и отладки в случаях, если расчёт итогового значения сложного арифметического выражения вызывает у них затруднения.

В блоке 3 – выполняется вывод итогового результата вычислений на переднюю панель ВП.

В блоке 4 – обозначено завершение вычислительного процесса, отрабатываемого ВП для поиска решения задачи, сформулированной в индивидуальной части задания для отработки навыков использования констант.

Стоит отметить, что при составлении итогового отчёта по данной работе настоятельно рекомендуется размещать в блоке (блоках) «Процесс» блок-схемы алгоритма математических форм записи арифметических выражений.

Напомним, что математическая форма записи может быть реализована при помощи [2]:

- встроенного в *Microsoft Office Word* редактора формул;
- дополнительного вставляемого объекта *Microsoft Equation 3.0*;
- дополнительно вставляемого объекта *Math Type* (требуется установки стороннего программного обеспечения).

### 3 Задание «Разработка ВП "Конвертор температуры"»

В пакете прикладных программ *National Instruments LabView* создать учебный ВП «Конвертор температуры» (или «Термометры»), моделирующий конвертацию температуры, заданной в градусах Цельсия, в другие единицы измерения. В рамках выполнения задания предусмотрена работа с такими единицами измерения температуры, как градусы:

- Кельвина,
- Фаренгейта,
- Ранкина,
- Реомюра,
- Рёмера,
- Ньютона,
- Делиля.

От обучающихся требуется настройка шкал деления всех виртуальных термометров (как в общей, так и в индивидуальной части задания) таким образом, чтобы предельные значения шкалы основного термометра, задающего градусы Цельсия, **по графическому уровню** совпадали с предельными значениями дополнительных термометров, показывающих иные единицы измерения.

Для повышения индивидуальности выполняемых обучающимися ВП диапазоны шкал термометров в градусах Цельсия задаются согласно Таблице 3.4.1, представленной в разделе 3.4, соответственно.

**Внимание!** Для случаев, когда стандартных размеров (габаритов) виртуального термометра оказывается недостаточно для нормального отображения засечек шкалы заданного диапазона, необходимо принудительно увеличить размеры термометра посредством использования инструмента позиционирования элементов управления *NI LabView* (*Position/Size/Select*).

Любой ВП потенциально является лабораторной установкой / лабораторным стендом. В связи с этим следует взять за правило, что он должен содержать на передней панели заголовок, авторство и год выпуска.

В процессе проектирования ВП необходимо решить вопрос о рациональном размещении функциональных элементов, как на графическом пользовательском интерфейсе – передней панели, так и на блок-диаграмме, содержащей исполняемый код.

Абсолютно всё, передаваемое в распоряжение конечного пользователя (или другого разработчика) ВП, должно быть читаемым, обладать минимальным количеством пересечений линий связи, а также быть понятным не только разработчику ВП, но и самому-самому неопытному пользователю программного обеспечения.

На блок-диаграмме все функциональные (то есть значащие) пересечения линий связи (узлы) обозначить точками. Для этого в меню перейти к настройкам «*Tools > Options...*», далее в выпадающем списке перейти к настройкам блок-диаграммы (*Block Diagram*) и выставить галочку напротив пункта «*Show dots at wire junctions*».

Правила конвертации одних температурных величин в другие представлены в разделе 3.2 («Полезные соотношения для выполнения задания»).

По итогам выполнения работы на проверку по электронной почте сдаются строго три файла (файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3):

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (\*.doc или \*.docx) с учётом рекомендаций, собранных в Приложении 2;
- файл виртуального прибора *National Instruments LabView* (\*.vi) по общей части работы;
- файл виртуального прибора *National Instruments LabView* (\*.vi) по индивидуальной части работы.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

В программе «Учебной практики» заданию присвоен номер «10».

### 3.1 Цель работы

Ознакомление с графическим пользовательским интерфейсом пакета прикладных программ *National Instruments LabView*, с его элементами, настройками и арифметическими операциями. Закрепление навыков составления блок-схем алгоритмов к авторским прикладным программам с семантическим (по смыслу) разбиением этих блок-схем по процедурам и функциям. Закрепление навыков математического вывода одних единиц измерения через другие.

### 3.2 Полезные соотношения для выполнения задания

#### 3.2.1 Формулы перевода температурных единиц

1. Перевод из градусов Цельсия в градусы Кельвина [14]:

$$K = 273,15 + X^{\circ}C ; \quad (3.2.1.1)$$

2.1. Перевод из градусов Фаренгейта в градусы Цельсия [15]:

$$T_{oC} = (t_F - 32) \cdot \frac{5}{9} ; \quad (3.2.1.2)$$

2.2. Перевод из градусов Цельсия в градусы Фаренгейта [15]:

$$t_F = T_{oC} \cdot \frac{9}{5} + 32 ; \quad (3.2.1.3)$$

3. Соответствие градусов Цельсия градусам Ранкина [16]:

$$1^{\circ}C = 493,47^{\circ}R ; \quad (3.2.1.4)$$

4. Соответствие градусов Цельсия градусам Реомюра [17]:

$$1^{\circ} C = 0,8^{\circ} Re'; \quad (3.2.1.5)$$

5. Перевод из шкалы Рёмера в градусы Цельсия [18]:

$$1^{\circ} C = \frac{40}{21} \cdot (1^{\circ} R_{\varnothing} - 7,5); \quad (3.2.1.6)$$

6. Соответствие градусов Цельсия градусам Ньютона [19]:

$$1^{\circ} C = \frac{100}{33} \cdot 1^{\circ} N; \quad (3.2.1.7)$$

7. Перевод из градусов Цельсия в градусы Делиля [20]:

$$T_{oD} = (100 - T_{oC}) \cdot \frac{3}{2}. \quad (3.2.1.8)$$

### 3.2.2 Приведение шкалы к единому графическому уровню

Рассмотрим задачу градуировки [21] виртуального термометра. Переместим значение управляющего воздействия «Термометра» к нижнему уровню. Хорошо видно, что на втором термометре, «Термометре 2» при этом получено значение, равное «100» единицам. Достигнута в точности верхняя граница.

Для данного примера попадание в точности в верхнюю границу результата при нижней границе исходных данных – это везение. В этом случае верхнюю границу менять не требуется (Рисунок 3.2.2.1).

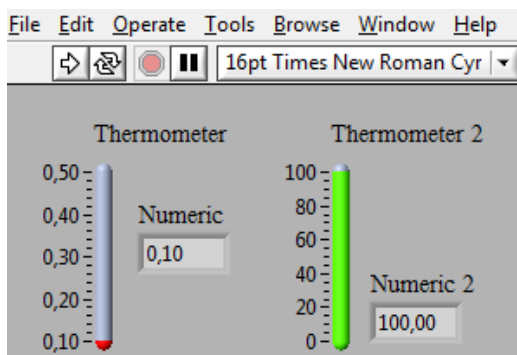


Рисунок 3.2.2.1 – Проверка нижнего уровня диапазона по активному параметру – прибор не зашкаливает. Это совпадение (случайность). Не требуется вносить какие-либо изменения

Теперь рассмотрим поведение интерфейсных элементов ВП при выставлении управляющего сигнала к верхней границе. Хорошо видно, что на втором, зелёном термометре внизу остался участок шкалы, который никогда не будет задействован при заданном диапазоне входного сигнала (задающего термометра). Так удалось выявить наличие избыточности на результирующей шкале (Рисунок 3.2.2.2). Любую избыточность необходимо устранять.

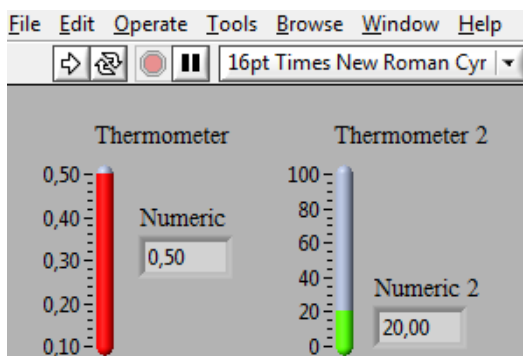


Рисунок 3.2.2.2 – Проверка верхнего уровня диапазона по активному параметру



Выполненная корректировка (Рисунок 3.2.2.3) гарантирует отсутствие зашкаливания, а также предоставляет избыточность проводимых измерений.

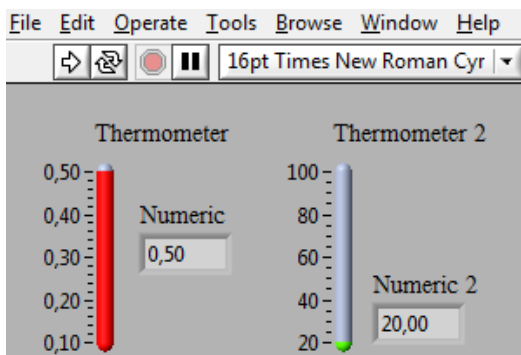


Рисунок 3.2.2.3 – Налажено соответствие шкал – иначе выполнена градуировка ВП

### 3.2.3 Инвертирование заполнения шкалы

Для особо пытливых инженерных умов в *National Instruments LabView* присутствует возможность настройки заполнителя шкалы («*Fill Options*»). Потому если настроить заполнение к максимуму («*Fill To Maximum*»), показанное на Рисунке 3.2.3.1, можно добиться полезного визуального эффекта.

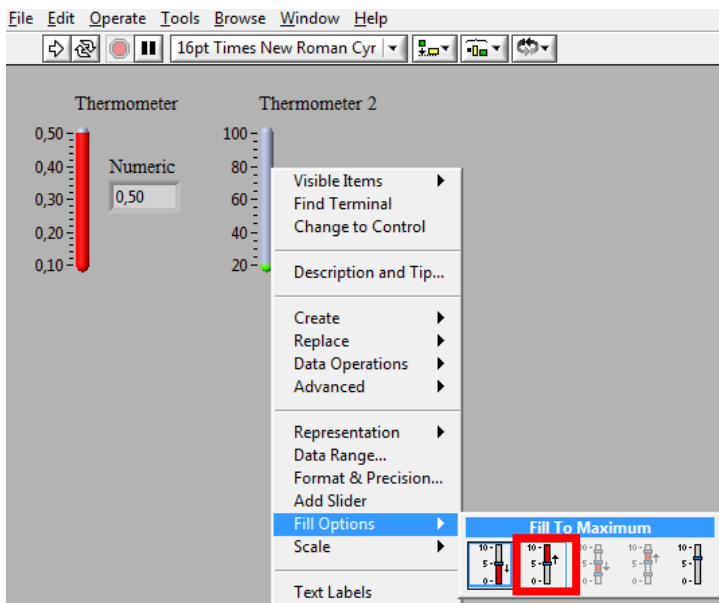


Рисунок 3.2.3.1 – Настройка обратной пропорциональной зависимости при заполнении шкал виртуальных термометров

Получаются идентичные по заполнителю термометры, объединённые обратной пропорциональной зависимостью (Рисунок 3.2.3.2).

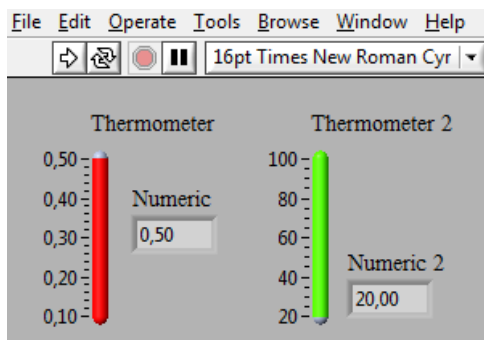


Рисунок 3.2.3.2 – Результат градуировки шкал виртуальных термометров с обратной пропорцией

### 3.2.4 Настройка начального значения задающего элемента управления

Часто для удобства работы с ВП требуется установка начальных приближений в интерфейсных элементах управления. Начальное приближение – то значение, которое демонстрируется на передней панели ВП по его загрузке. Для числовых элементов управления это, как правило «0» («ноль»).

Выполнять настройку начального приближения удобно при отладке ВП, чтобы каждый раз при повторном открытии файла, содержащего ВП, не задавать одно и то же, нужное для тестирования значение. Этот подход существенно экономит время разработчика.

Перейти к настройке начального приближения для интерфейсного элемента управления можно через его контекстное меню, как показано на Рисунке 3.2.4.1.

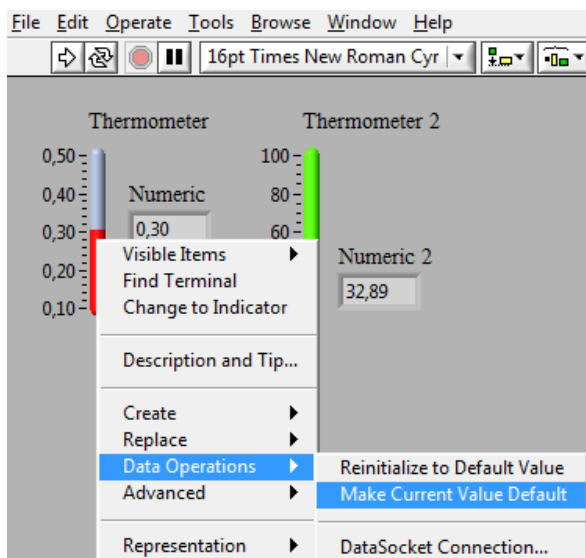


Рисунок 3.2.4.1 – Контекстное меню красного термометра. Фиксация выставленного значения заданным по умолчанию (*Make Current Value Default*)

Результат открытия файла ВП после изменения значения, установленного по умолчанию, показано на Рисунке 3.2.4.2. Все значения сброшены в минимум или ноль, кроме значения красного термометра.

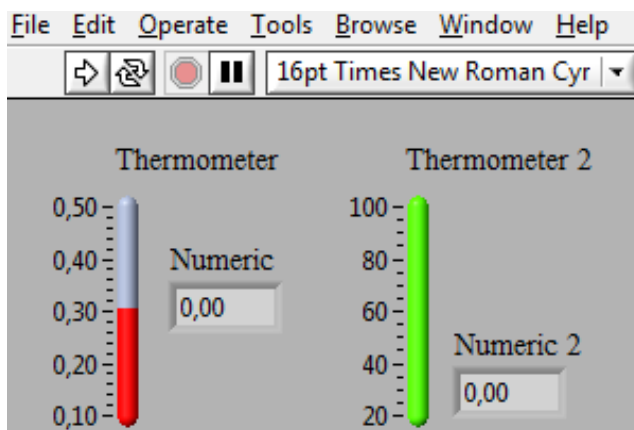


Рисунок 3.2.4.2 – Результат отображения ВП после повторного открытия *vi*-файла с созданным ВП, в котором изменено начальное приближение

Представленных в разделе 3.2 сведений в сочетании с ранее приобретёнными навыками проектирования ВП по итогам выполнения задания номер «9» достаточно для выполнения обучающимися поставленной задачи номер «10».

### 3.3 Пример выполнения задания

#### 3.3.1 Смена режима с элемента вывода на элемент ввода

Элемент управления «термометр» в пакете прикладных программ *National Instruments LabView* размещается разделе числовых элементов управления (*Numeric*). Его размещение на панели элементов управления отмечено на Рисунке 3.3.1.1.

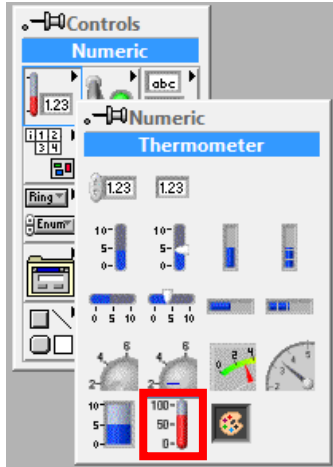


Рисунок 3.3.1.1 – Демонстрация с маркировкой места расположения термометра в разделе числовых интерфейсных элементов управления («*Numeric Controls*»)

По умолчанию шкала стандартного термометра выставлена в диапазоне от «0» до «100» (Рисунок 3.3.1.2).

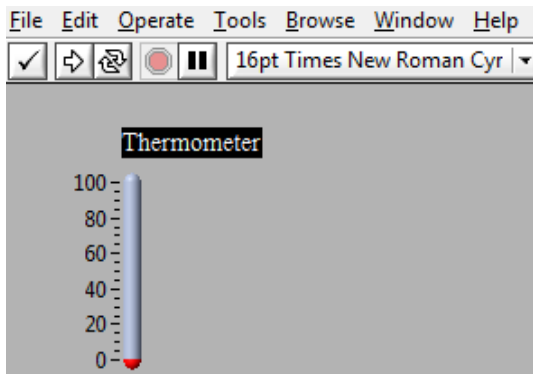


Рисунок 3.3.1.2 – Начальное представление интерфейсного элемента «Термометр» (*Thermometer*)

На блок-диаграмме термометр по умолчанию – это элемент индикации (вывода информации) (Рисунок 3.3.1.3). Данная настройка логична и естественна. В жизни термометры отображают значения температуры, например, окружающей среды. В *NI LabView* разработчику и конечному пользователю предоставлена возможность изменить привычное и естественное поведение термометра – организовывать управление при помощи виртуального термометра.

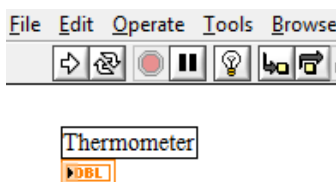


Рисунок 3.3.1.3 – По умолчанию на блок-диаграмме термометр является элементом вывода информации (подключение слева)

Для перевода виртуального термометра в режим элемента управления необходимо нажать правой кнопкой мыши на соответствующий ему прямоугольный элемент на блок-диаграмме. Будет вызвано контекстное меню (Рисунок 3.3.1.4), в котором следует выбрать позицию «Изменить на элемент ввода» (*«Change to Control»*).

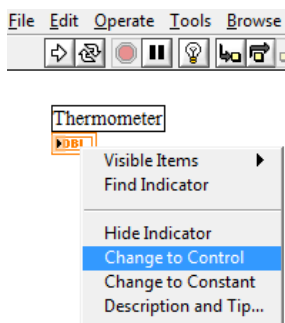


Рисунок 3.3.1.4 – Смена «полярности» виртуального термометра с элемента вывода на элемент ввода

В результате выполненных действий образ элемента на блок-диаграмме изменит «полярность» – станет элементом ввода (подключение справа) (Рисунок 3.3.1.5).

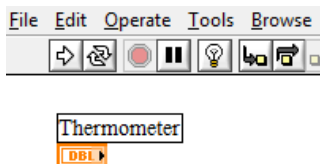


Рисунок 3.3.1.5 – Виртуальный управляющий (входной) термометр на блок-диаграмме

Для удобства отслеживания изменений на передней панели ВП к полученному элементу ввода следует подключить стандартный числовой элемент вывода (Рисунок 3.3.1.6).

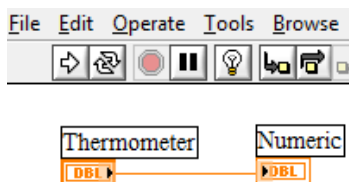


Рисунок 3.3.1.6 – Управляющий (входной) термометр, однозначно связанный со стандартным числовым элементом вывода

Далее на передней панели ВП посредством использования инструмента для изменения значения («*Operate Value*») сместим заполнитель термометра к середине диапазона и запустим ВП в режиме однократного исполнения («*Run*»). Результат данной манипуляции изображён на Рисунке 3.3.1.7.

Благодаря стандартному числовому элементу вывода при отсутствии достаточной детализации шкалы виртуального термометра, можно получать вывод более точных значений заданной температуры. Точность при этом зависит от настройки формата вывода данных на соответствующий элемент вывода.

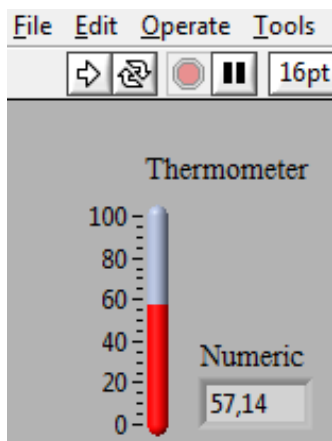


Рисунок 3.3.1.7 – Результат запуска ВП «Задающий термометр с выводом информации» в режиме разового исполнения

Существует более простой и быстрый способ численно индицировать точные показания виртуального термометра. Можно не вводить отдельно стоящий числовой элемент вывода, а воспользоваться внутренним. Для этого перейти в контекстное меню виртуального термометра, щёлкнув по нему правой кнопкой мыши и последовательно выбрать «Отображаемые элементы» > «Цифровой дисплей» («*Visible Items*» > «*Digital Display*») так, как показано на Рисунке 3.3.1.8.

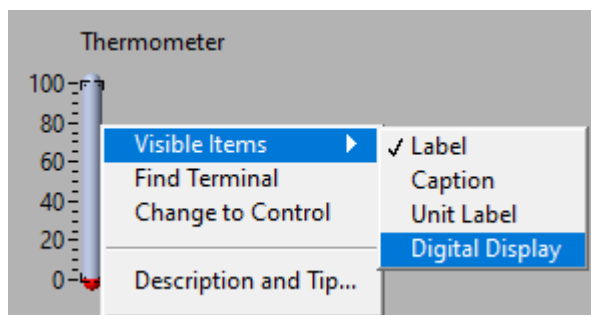


Рисунок 3.3.1.8 – Активация внутреннего числового элемента вывода, исходно связанного с виртуальным термометром



### 3.3.2 Изменение диапазона шкалы виртуального термометра

Для большинства разрабатываемых в *NI LabView* ВП стандартного диапазона шкалы виртуального термометра бывает недостаточно для решения поставленных задач.

Покажем, что изменение диапазона шкалы виртуального термометра не является сложной задачей. Она решается с использованием инструмента редактирования текста («*Edit Text*»).

Выберем при активированном инструменте редактирования текста нижнее значение диапазона шкалы термометра и заменим его вводом числовых значений с клавиатуры, например, с нуля на «-50» (Рисунок 3.3.2.1).

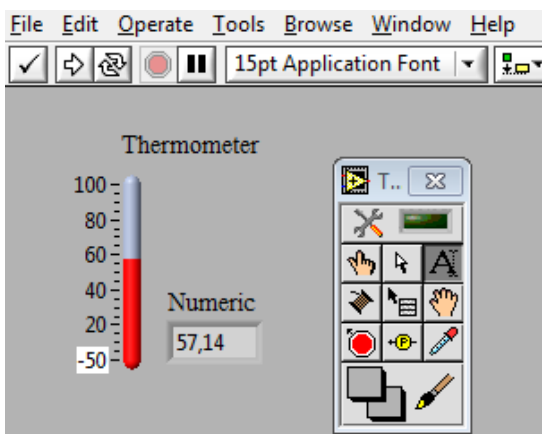


Рисунок 3.3.2.1 – Демонстрация изменения значения нижней границы диапазона термометра с использованием инструмента редактирования текста («*Edit Text*»)

После подтверждения выполненного изменения посредством нажатия на клавишу «*Enter*» на клавиатуре изменённое значение нижней границы диапазона шкалы виртуального термометра вступит в силу (Рисунок 3.3.2.2).

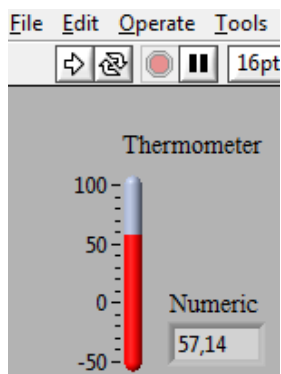


Рисунок 3.3.2.2 – Результат изменения нижней границы диапазона шкалы термометра

### 3.3.3 Переход к прецизионным измерениям

Часто у начинающих разработчиков ВП возникает проблема с тем, чтобы задать виртуальному термометру шкалу с дробными значениями в *National Instruments LabView*.

Разберём соответствующую настройку на примере редактирования шкалы прецизионного термометра, работающего в диапазоне от нуля до одного градуса (Рисунок 3.3.3.1).

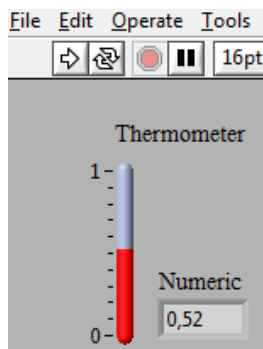


Рисунок 3.3.3.1 – Прецизионный термометр

Эта задача имеет несложное решение. Для начала необходимо обратиться к контекстному меню термометра и в разделе шкала («*Scale*») перейти к диалоговому окну, скрывающемуся за пунктом «Формат и Точность» («*Format & Precision*»). Переход по контекстному меню показан на Рисунке 3.3.3.2.

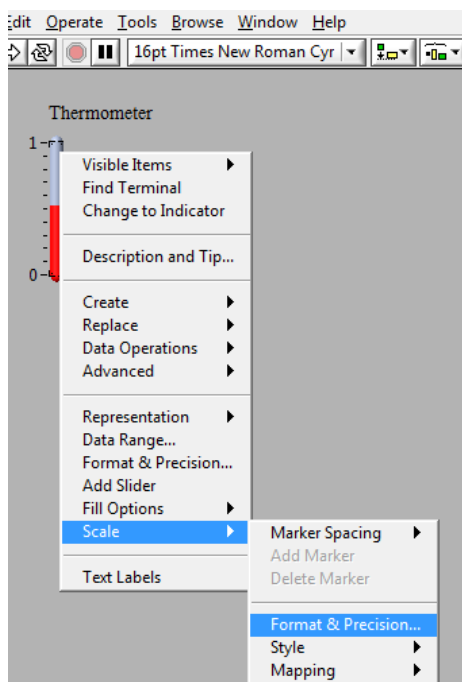


Рисунок 3.3.3.2 – В контекстном меню термометра показан раздел настройки шкалы («*Scale*») с отметками переходов к диалоговому окну «Формат и Точность» («*Format & Precision*»)

В открывшемся диалоговом окне необходимо выставить требуемую точность по количеству знаков после плавающей запятой («*Digits of Precision*»). В данном примере рассматривается точность с двумя знаками после плавающей запятой (Рисунок 3.3.3.3).

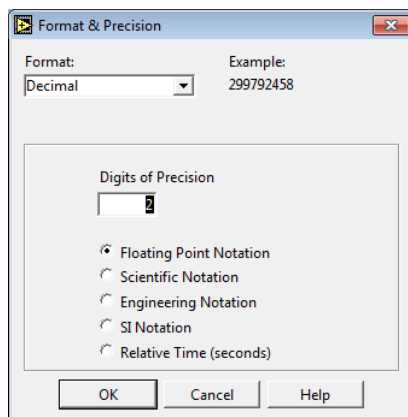


Рисунок 3.3.3.3 – Диалоговое окно «Формат и Точность» с изменённым значением точности до двух знаков после запятой

После изменения значения с «0» на «2» нажатием на кнопку «OK». Шкала виртуального термометра изменится. На ней появятся промежуточные вещественные значения. Выполненная настройка также откроет возможность для редактирования цифр, расположенных справа от плавающей запятой при изменении границ диапазона шкалы (Рисунок 3.3.3.4). Предыдущие настройки формата шкалы термометра не позволили бы разработчику выполнить ввод и корректировку вещественной части числовых значений.

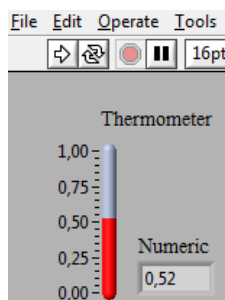


Рисунок 3.3.3.4 – Демонстрация шкалы, учитывающей два знака после плавающей запятой

Как только мы отказались от стандартной точности шкалы с округлением до ближайшего целого значения, у нас появилась возможность для ещё большего сокращения диапазона шкалы виртуального термометра.

Такой элемент управления (Рисунок 3.3.3.5) позволяет в полном объёме исследовать, например, обратно пропорциональные температурные зависимости.

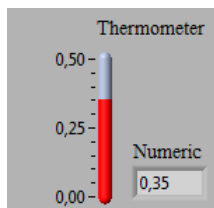


Рисунок 3.3.3.5 – Маломасштабный / точный / прецизионный виртуальный термометр

### 3.3.4 Программирование простейшей зависимости в обратной пропорции

Для реализации упомянутого в подразделе 3.3.3 исследования обратной пропорциональной зависимости на графическом пользовательском интерфейсе ВП понадобится разместить второй виртуальный термометр (Рисунок 3.3.4.1).

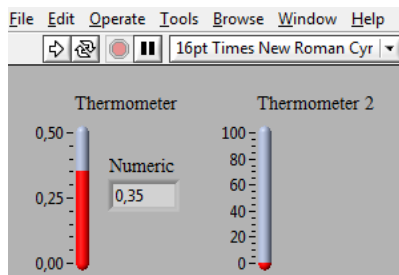


Рисунок 3.3.4.1 – Дополнение передней панели ВП «Прецизионный термометр» вторым термометром

На Рисунке 3.3.4.2 показан графический код, размещённый на блок-диаграмме ВП и позволяющий организовать обратную пропорциональную зависимость «10 / C°».

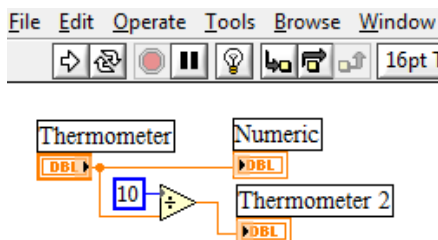


Рисунок 3.3.4.2 – Обратная пропорциональная зависимость

Так большему входному значению соответствует меньшее выходное значение (Рисунок 3.3.4.3).

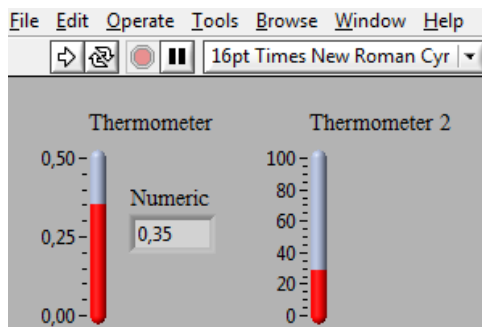


Рисунок 3.3.4.3 – Результат работы программы конвертации единиц измерения температуры

### 3.3.5 Настройка цветового оформления виртуального термометра

Разграничим управляющий (входной) термометр и индицирующий (выходной) термометр цветом (это важно знать для выполнения индивидуальной части задания).

Переход к изменению цветовых настроек элементов передней панели ВП показан на Рисунке 3.3.5.1.

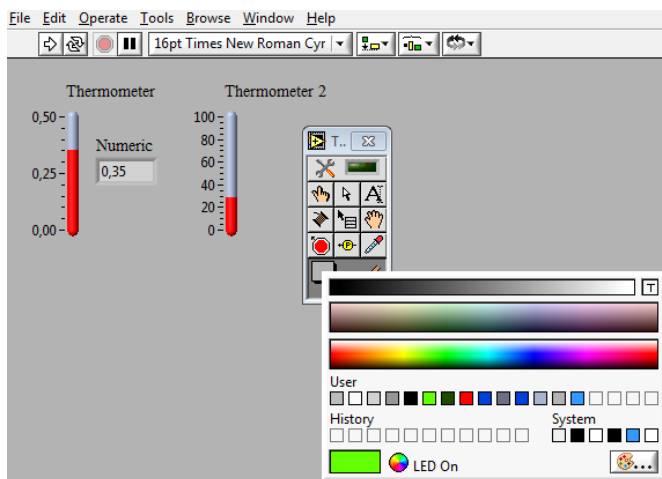


Рисунок 3.3.5.1 – Изменение активного цвета для инструмента окрашивания верхних слоёв элементов управления

После выбора заполнителя термометра при активном инструменте установки цвета («Set Color»), заполнитель изменит цвет на зелёный (Рисунок 3.3.5.2).

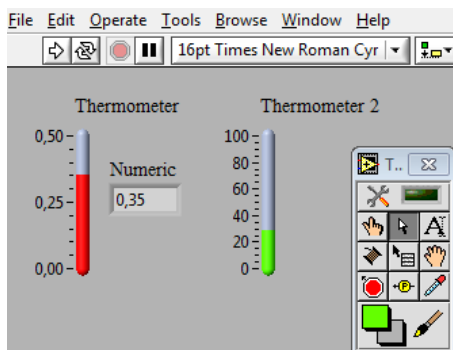


Рисунок 3.3.5.2 – Изменён цвет индицирующего (выходного) термометра

### 3.4 Варианты индивидуального задания

**Особые требования, предъявляемые к ВП в зависимости от чётности / нечётности варианта обучающегося в списке группы:**

- чётные варианты как в общей, так и в индивидуальной части задания используют в качестве задающего элемента управления входной термометр опорной измерительной системы (в общей части задания опорной системой измерения для всех обучающихся являются градусы Цельсия);
- нечётные варианты используют в качестве задающего элемента управления – стандартный числовой элемент ввода опорной системы измерения (все термометры ВП – элементы вывода).

Величины предельных значений каждого термометра могут и должны отличаться друг от друга. Например, если для градусов Цельсия задан диапазон от -100 до 100, то для градусов Кельвина должен быть задан диапазон от 173,15 до 373,15.

Таблица 3.4.1 – Диапазоны шкал задающих виртуальных термометров (как в общей, так и в индивидуальной части)

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
130...230	2.9...6.1	-1...99	0.12...3.45	-1000...1000
<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
5...55	100...150	-70.77...15.58	0...10	-50...0
<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
0...80	220...330	-500...300	-15...15	-20...-10
<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>
205...215	-5...55	-45...10	-0.9...0.1	0...100
<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>
300...500	-22...22	-10000...10	0.001...0.1	-0.001...0
<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
1...101	0.9...30.9	0.0001...1	500...650	-23.4...5.67
<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>
-30...30	0.1...0.9	12.77...40.92	-78.9...-12.3	0...4



Выполнить второй ВП для исследования конвертации температур, в котором за основу взять единицы измерения, заданные по варианту.

Таблица 3.4.2 – Содержание индивидуальной части задания согласно номеру варианта обучающегося в списке группы

<b>№ вар.</b>	<b>Содержание задания</b>
1.	Выполнить прямой перевод из градусов Кельвина в градусы Делиля. Цвет входного термометра оранжевый; выходного – фиолетовый.
2.	Выполнить прямой перевод из градусов Фаренгейта в градусы Реомюра. Цвет входного термометра зелёный; выходного – красный.
3.	Выполнить прямой перевод из градусов Делиля в градусы Кельвина. Оба термометра – оливкового цвета.
4.	Выполнить прямой перевод из градусов Ньютона в градусы Реомюра. Оба термометра – фиолетового цвета.
5.	Выполнить прямой перевод из градусов Фаренгейта в градусы Рёмера. Цвет входного термометра серый; выходного – розовый.
6.	Выполнить прямой перевод из градусов Реомюра в градусы Рёмера. Оба термометра – зелёного цвета.
7.	Выполнить прямой перевод из градусов Кельвина в градусы Ранкина. Цвет входного термометра синий; выходного – зелёный.
8.	Выполнить прямой перевод из градусов Делиля в градусы Цельсия. Оба термометра – серые.

<b>№ вар.</b>	<b>Содержание задания</b>
9.	Выполнить прямой перевод из градусов Фаренгейта в градусы Ранкина. Цвет входного термометра фиолетовый; выходного – синий.
10.	Выполнить прямой перевод из градусов Ньютона в градусы Кельвина. Цвет входного термометра салатовый; выходного – алый.
11.	Выполнить прямой перевод из градусов Делиля в градусы Реомюра. Цвет входного термометра голубой; выходного – серый.
12.	Выполнить прямой перевод из градусов Кельвина в градусы Рёмера. Цвет входного термометра чёрный; выходного – салатовый.
13.	Выполнить прямой перевод из градусов Ньютона в градусы Делиля. Цвет входного термометра бирюзовый; выходного – лайм.
14.	Выполнить прямой перевод из градусов Фаренгейта в градусы Цельсия. Цвет входного термометра синий; выходного – красный.
15.	Выполнить прямой перевод из градусов Делиля в градусы Рёмера. Цвет входного термометра красный; выходного – коричневый.
16.	Выполнить прямой перевод из градусов Кельвина в градусы Реомюра. Цвет входного термометра розовый; выходного – салатовый.
17.	Выполнить прямой перевод из градусов Делиля в градусы Ранкина. Цвет входного термометра чёрный; выходного – красный.

<b>№ вар.</b>	<b>Содержание задания</b>
18.	Выполнить прямой перевод из градусов Фаренгейта в градусы Кельвина. Цвет входного термометра жёлтый; выходного – белый.
19.	Выполнить прямой перевод из градусов Ньютона в градусы Цельсия. Цвет морской волны для входного термометра; выходного – красный.
20.	Выполнить прямой перевод из градусов Делиля в градусы Фаренгейта. Цвет входного термометра голубой; выходного – оранжевый.
21.	Выполнить прямой перевод из градусов Кельвина в градусы Цельсия. Цвет входного термометра белый; выходного – чёрный.
22.	Выполнить прямой перевод из градусов Рёмера в градусы Реомюра. Цвет входного термометра синий; выходного – жёлтый.
23.	Выполнить прямой перевод из градусов Фаренгейта в градусы Делиля. Оба термометра – салатового цвета.
24.	Выполнить прямой перевод из градусов Реомюра в градусы Ньютона. Оба термометра – оранжевого цвета.
25.	Выполнить прямой перевод из градусов Кельвина в градусы Ньютона. Цвет входного термометра оливковый; выходного – оранжевый.
26.	Выполнить прямой перевод из градусов Фаренгейта в градусы Ньютона. Цвет входного термометра синий; выходного – жёлтый.
27.	Выполнить прямой перевод из градусов Рёмера в градусы Цельсия. Цвет входного термометра серый; выходного – розовый.

<b>№ вар.</b>	<b>Содержание задания</b>
28.	Выполнить прямой перевод из градусов Делиля в градусы Ньютона. Оба термометра – белого цвета.
29.	Выполнить прямой перевод из градусов Кельвина в градусы Фаренгейта. Цвет входного термометра красный; выходного – синий.
30.	Выполнить прямой перевод из градусов Ньютона в градусы Рёмера. Цвет входного термометра алый; выходного – бледно-зелёный.
31.	Выполнить прямой перевод из градусов Реомюра в градусы Цельсия. Цвет входного термометра золотой; выходного – бронзовый.
32.	Выполнить прямой перевод из градусов Ньютона в градусы Фаренгейта. Цвет входного термометра фуксия; выходного – фишашковый.
33.	Выполнить прямой перевод из градусов Ранкина в градусы Кельвина. Цвет входного термометра индиго; выходного – периковый.
34.	Выполнить прямой перевод из градусов Реомюра в градусы Фаренгейта. Цвет входного термометра хаки; выходного – лазурный.
35.	Выполнить прямой перевод из градусов Ньютона в градусы Ранкина. Оба термометра – лилового цвета.

### 3.5 Пример компоновки графического пользовательского интерфейса ВП

На Рисунке 3.5.1 представлен фрагмент передней панели ВП, который необходимо реализовать в общей части задания. Термометры можно размещать в один ряд как вертикально, так и горизонтально, можно предусмотреть компоновку в несколько рядов.

Термометр в градусах Цельсия является управляющим (входным), все остальные элементы – управляемые (индикаторы / элементы вывода). Цвет заполнителя термометров в общем случае в общей части задания можно не изменять.

Маркер «градус» («°») выполнен отдельно от латинского наименования величины единиц измерения температуры. Текстом меньшего размера на передней панели ВП выполнена строчная буква «о».

Шкала управляющего элемента размечается согласно Таблице 3.4.1, управляемых элементов – по результатам расчётов граничных значений для каждой системы единиц.

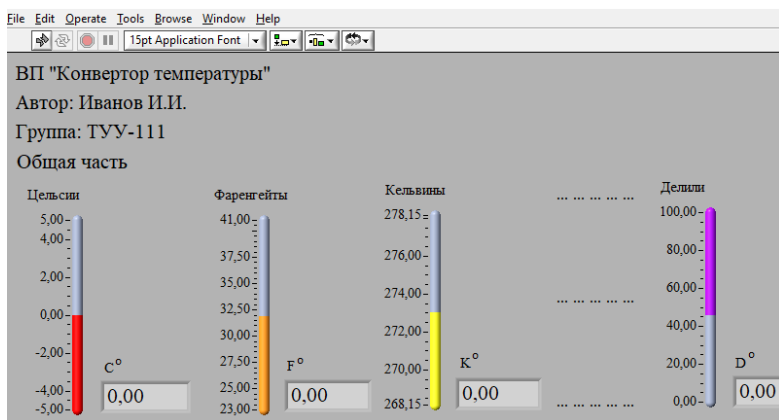


Рисунок 3.5.1 – Опорный пример передней панели ВП, создаваемого в общей части задания

На Рисунке 3.5.2 представлен образец компоновки передней панели ВП, который необходимо разработать в индивидуальной части задания. Вместо надписей: «Входная система измерения» и «Выходная система измерения» должны быть указаны соответствующие наименования, например, «Фаренгейты» и «Ньютоны». Аналогично, вместо  $T1$  и  $T2$ , должно быть указано конкретное обозначение. Для выбранного примера это « $F$ » и « $N$ », соответственно.

Отчёт по работе должен содержать вывод формулы для конвертации единиц. Формула не обязательно должна быть найдена в справочнике или в сети Интернет, её можно вывести пересчётом через градусы Цельсия.

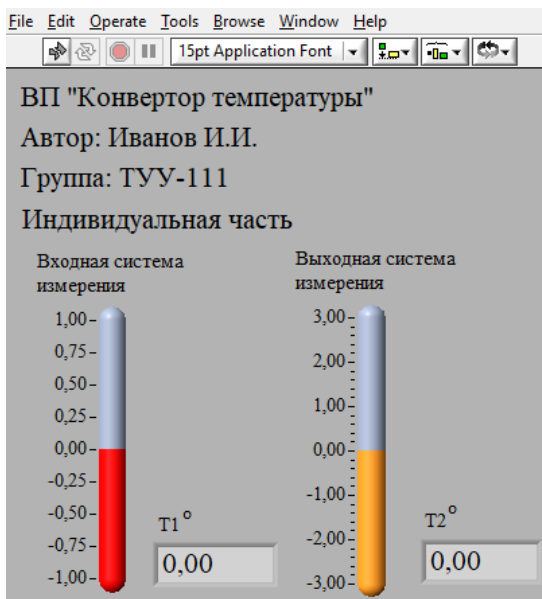


Рисунок 3.5.2 – Шаблон передней панели ВП, создаваемого в индивидуальной части задания

### 3.6 Рекомендации по размещению информации на блок-диаграмме ВП

На Рисунке 3.6.1 выполнена визуализация компоновки элементов на блок-диаграмме ВП по разделам: «Входная информация», «Константы», «Операции» и «Выходная информация». Можно взять эту схему за основу, но после отладки ВП не забыть выполнить минимизацию линий связи. Здесь это требование намеренно не соблюдается, чтобы сделать прочтение схемы более комфортным для начинающих разработчиков.

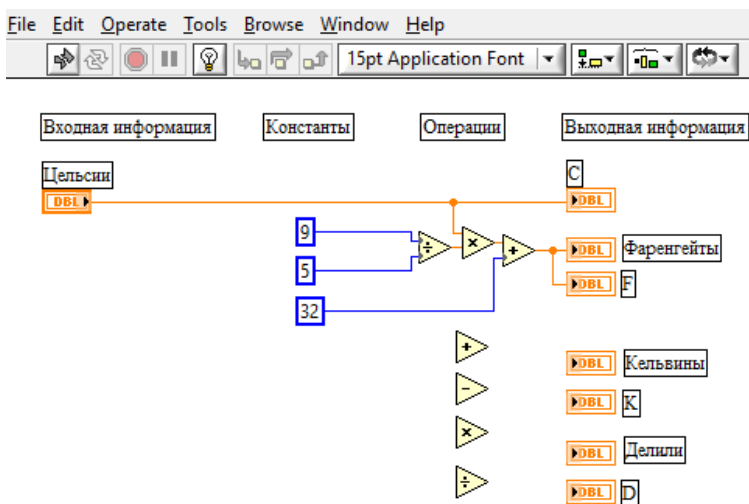


Рисунок 3.6.1 – Визуализация рекомендаций по начальному размещению элементов на блок-диаграмме ВП, разрабатываемого в общей части задания. Соединена схема конвертации в градусы Фаренгейта

На Рисунке 3.6.2 выполнена шаблонная компоновка элементов на блок-диаграмме ВП, разрабатываемого в индивидуальной части задания. Обучающимся важно понять,

что ВП в индивидуальной части в любом случае получится проще и компактнее, чем ВП из общей части.

Сложность и особенность индивидуальной части задания состоит в выводе специфической формулы конвертации температур и в отработке навыков настройки цветового оформления интерфейсных элементов управления, размещаемых на передней панели ВП.

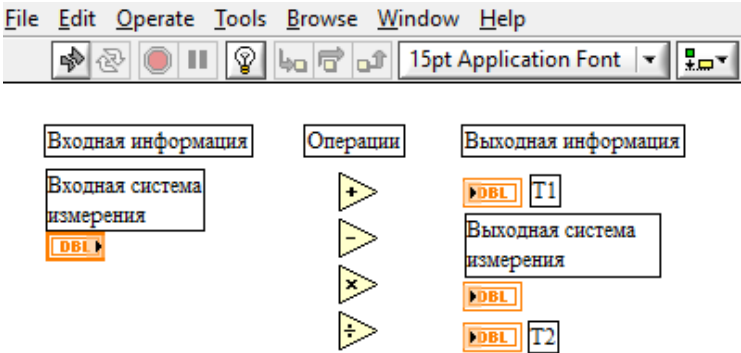


Рисунок 3.6.2 – Визуализация рекомендаций по начальному размещению элементов на блок-диаграмме ВП, разрабатываемого в индивидуальной части задания



### 3.7 Блок-схемы алгоритмов для типовых решений

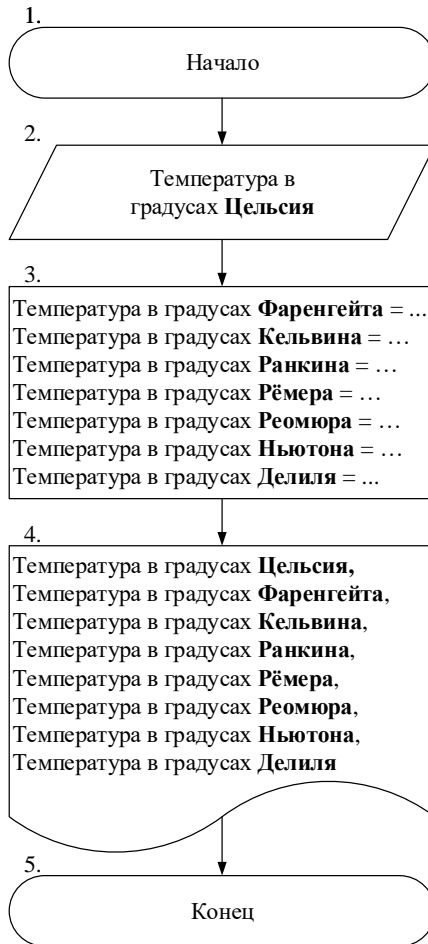


Рисунок 3.7.1 – Схема алгоритма для ВП из общей части задания

Описание алгоритма, представленного на Рисунке 3.7.1:

В блоке 1 – обозначено начало вычислительного процесса, обрабатываемого ВП для поиска значений в задаче, сформулированной в общей части задания.

В блоке 2 – указаны исходные данные ВП для конвертации значений температуры – температура, заданная в градусах Цельсия.

В блоке 3 – выполняется конвертация температуры по представленным зависимостям (на Рисунке 3.7.1 эти зависимости отсутствуют – обучающимся предлагается выполнить эти записи в рамках самостоятельной работы).

В блоке 4 – выполняется вывод рассчитанных в блоке 3 значений на соответствующие элементы вывода, размещённые на передней панели ВП.

В блоке 5 – обозначено завершение вычислительного процесса, обрабатываемого ВП для поиска значений в задаче, сформулированной в общей части задания.



Рисунок 3.7.2 – Схема алгоритма для ВП из индивидуальной части задания

### Описание алгоритма, представленного на Рисунке 3.7.2:

В блоке 1 – обозначено начало вычислительного процесса, обрабатываемого ВП для поиска значений в задаче, сформулированной в индивидуальной части задания.

В блоке 2 – указаны исходные данные ВП для конвертации значений температуры – температура, заданная в градусах системы, указанной по варианту согласно Таблице 3.4.2 как «входная».

В блоке 3 – выполняется конвертация температуры согласно выведенной и представленной зависимости (на Рисунке 3.7.2 представлена обобщённая формула, основанная на допущении, что любая зависимость в данном задании будет получаться линейной; для каждого варианта задания – это будет своя зависимость со своими значениями коэффициентов конвертации).

В блоке 4 – выполняется вывод рассчитанной в блоке 3 температуры в градусах системы, указанной по варианту согласно Таблице 3.4.2 как «выходная».

В блоке 5 – обозначено завершение вычислительного процесса, обрабатываемого ВП для поиска значений в задаче, сформулированной в индивидуальной части задания.

#### 4      Задание «Разработка ВП "Умный калькулятор (CASE-калькулятор)"»

В пакете прикладных программ *National Instruments LabView* разработать ВП «Умный калькулятор», в состав которого входят операции, заданные согласно Таблице 4.4.1.

Операции должны быть размещены на графическом пользовательском интерфейсе строго в том порядке, в котором они упоминаются в Таблице 4.4.1.

Определение именно той операции, которую пожелал реализовать пользователь, должно выполняться в соответствии с некоторой логикой, задаваемой булевыми элементами управления, например, кнопками или тумблерами.

В процессе разработки графического пользовательского интерфейса предусмотреть:

- числовые элементы ввода в необходимом количестве для ввода значений операндов,
- один единственный числовой элемент для вывода результата/ответа,
- логические элементы ввода в необходимом количестве для однозначного определения системой выполняемой операции.

Результат/ответ направлять на элемент вывода только при единственном включённом логическом элементе ввода.

Исключить одновременное включение нескольких логических элементов ввода и либо не реагировать на подобный входной сигнал – оставлять на выходе ноль («NaN»), либо при включении нескольких элементов ввода выдавать ошибки, формируемые в виде числового кода, например, «9999», «8888», которые следует расшифровать для пользователя на графическом пользовательском интерфейсе дополнительной таблицей соответствия.

Далее приведены примеры расшифровок кодов, которые могут быть размещены на графическом пользовательском интерфейсе:

- «9999» – недопустимо одновременное умножение и деление,
- «8888» – недопустимо одновременное сложение и вычитание,
- «7777» – недопустимо деление на ноль,
- и так далее.

Разработчику ВП требуется продумать структуру других возможных ошибок самостоятельно.

На блок-диаграмме все функциональные, то есть значащие пересечения линий связи (узлы) обозначить точками. Для этого в меню перейти к настройкам «*Tools > Options...*», далее в выпадающем списке перейти к настройкам блок-диаграммы (*Block Diagram*) и выставить галочку напротив пункта «*Show dots at wire junctions*».

По итогам выполнения работы на проверку по электронной почте сдаются строго два файла (файлы, направляемые на проверку по электронной почте, должны быть названы в соответствии с требованиями, собранными в Приложении 3):

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (\*.doc или \*.docx) с учётом рекомендаций, собранных в Приложении 2;
- файл виртуального прибора *National Instruments LabView* (\*.vi) по индивидуальной части работы.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

Необходимо продумать индивидуальность подхода к решению логического наполнения «Умного калькулятора» на

базе CASE-структуры. Двух одинаковых работ в студенческой группе быть не должно.

С целью повышения индивидуальности выполняемой работы допускается реализация ВП на базе функции выбора («*Select*»), располагающейся в разделе элементов сравнения («*Comparison*»).

В программе «Учебной практики» заданию присвоен номер «11».

## **4.1 Цель работы**

Закрепление навыков разработки графических пользовательских интерфейсов ВП и составления графического кода на языке «*G*» в пакете прикладных программ *National Instruments LabView*. Приобретение навыков работы с CASE-структурами (одиночными / независимыми / вложенными), входящими в состав пакета прикладных программ *National Instruments LabView*. Закрепление изученного материала в курсе «Программирование и основы алгоритмизации» в разделе «Разветвляющийся вычислительный процесс».

## **4.2 Полезные соотношения для выполнения задания**

### **4.2.1 Основные числовые функции, размещаемые на блок-диаграммах ВП**

Раздел числовых элементов, размещаемых на блок-диаграммах ВП, содержит следующие функции языка «*G*», подразделы, а также константы, изображённые на Рисунке 4.2.1.1 (чтение ведётся слева направо, сверху вниз):

- операция сложения («*Add*»);
- операция вычитания («*Subtract*»);
- операция умножения («*Multiply*»);
- операция деления («*Divide*»);

- операции целочисленного деления и получения остатка от деления («*Quotient & Remainder*»);
- подраздел конвертации («*Conversion*»);
- операция инкрементирования / добавления единицы («*Increment*»);
- операция декрементирования / вычитания единицы («*Decrement*»);
- операция суммирования элементов массива («*Add Array Elements*»);
- операция перемножения элементов массива («*Multiply Array Elements*»);
- элемент, объединяющий арифметические операции («*Compound Arithmetic*»);
- подраздел тригонометрии («*Trigonometric*»);
- операция взятия абсолютной величины / модуля («*Absolute Value*»);
- операция математического округления до ближайшего целого («*Round To Nearest*»);
- операция округления в меньшую сторону («*Round To -Infinity*»);
- операция округления в большую сторону («*Round To +Infinity*»);
- операция выдачи реализации псевдослучайной величины в диапазоне от нуля до единицы («*Random Number (0-1)*»);
- подраздел логарифмов («*Logarithmic*»);
- операция взятия квадратного корня («*Square Root*»);
- операция инверсии / взятия значения с обратным знаком («*Negate*»);
- операция масштабирования по степени двойки / умножение на 2 в степени  $N$  («*Scale By Power Of 2*»);
- операция сигнум (определения знака): пришло любое положительное – в ответе «1», пришло любое отрицательное – в ответе «-1», пришёл ноль – в ответе «0» («*Sign*»);



- операция обращения / деления единицы на заданное значение («*Reciprocal*»);
- подраздел операций над комплексными числами («*Complex*»);
- числовая константа («*Numeric Constant*»);
- перечислимая константа («*Enum Constant*»);
- константа списка значений («*Ring Constant*»);
- узел математических выражений («*Expression Node*»);
- подраздел дополнительных констант («*Additional Numeric Constants*»).

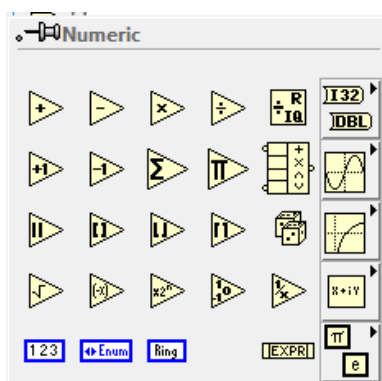


Рисунок 4.2.1.1 – Раздел числовых («*Numeric*») функций, размещаемых на блок-диаграммах ВП

Далее рассмотрен подраздел тригонометрических функций (Рисунок 4.2.1.2):

- синус («*Sine*»);
- косинус («*Cosine*»);
- тангенс («*Tangent*»);
- арксинус («*Inverse Sine*»);
- арккосинус («*Inverse Cosine*»);
- арктангенс («*Inverse Tangent*»);
- гиперболический синус («*Hyperbolic Sine*»);
- гиперболический косинус («*Hyperbolic Cosine*»);

- гиперболический тангенс («*Hyperbolic Tangent*»);
- гиперболический арксинус («*Inverse Hyperbolic Sine*»);
- гиперболический арккосинус («*Inverse Hyperbolic Cosine*»);
- гиперболический арктангенс («*Inverse Hyperbolic Tangent*»);
- косеканс («*Cosecant*»);
- секанс («*Secant*»);
- котангенс («*Cotangent*»);
- объединённых синуса и косинуса («*Sine & Cosine*»);
- арктангенс двух аргументов («*Inverse Tangent (2 Input)*»);
- отсчётов («*Sinc*»).

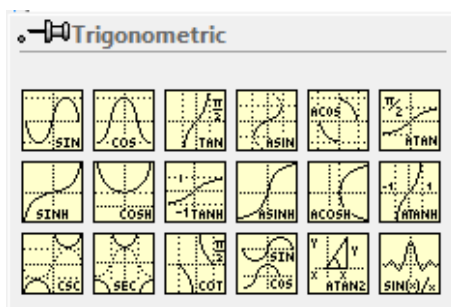


Рисунок 4.2.1.2 – Подраздел тригонометрических («*Trigonometric*») функций раздела численных («*Numeric*») функций, размещаемых на блок-диаграммах ВП

Далее рассмотрен подраздел логарифмических функций (Рисунок 4.2.1.3):

- экспонента («*Exponential*»);
- степень десяти («*Power Of 10*»);
- степень двойки («*Power Of 2*»);
- возведение  $X$  в степень  $Y$  («*Power Of X*»);
- экспонента, проходящая через начало координат («*Exponential (Arg) - 1*»);

- натуральный логарифм («*Natural Logarithm*»);
- логарифм по основанию десяти («*Logarithm Base 10*»);
- логарифм по основанию двойки («*Logarithm Base 2*»);
- логарифм по основанию  $X$  («*Logarithm Base X*»);
- натуральный логарифм, проходящий через начало координат («*Natural Logarithm (Arg + 1)*»).

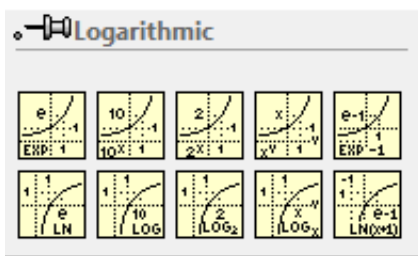


Рисунок 4.2.1.3 – Подраздел логарифмических («*Logarithmic*») функций раздела численных («*Numeric*») функций, размещаемых на блок-диаграммах ВП

Далее рассмотрен подраздел дополнительных численных констант (Рисунок 4.2.1.4):

- цветовая константа («*Color Box Constant*»);
- символьная константа списка значений для интерфейсного элемента «список» («*Listbox Symbol Ring Constant*»);
- константа списка ошибок («*Error Ring Constant*»);
- Пифагорова константа («*Pi*»);
- Пифагорова константа, умноженная на два («*Pi Multiplied By 2*»);
- Пифагорова константа, делённая на два («*Pi Divided By 2*»);
- обращённая Пифагорова константа («*Reciprocal Of Pi*»);
- натуральный логарифм Пифагоровой константы («*Natural Logarithm Of Pi*»);
- минус бесконечность («*Negative Infinity*»);

- основание натурального логарифма («*Natural Logarithm Base*»);
- обращённое основание натурального логарифма («*Reciprocal Of e*»);
- логарифм экспоненты по основанию десяти («*Base 10 Logarithm Of e*»);
- натуральный логарифм десяти («*Natural Logarithm Of 10*»);
- натуральный логарифм двойки («*Natural Logarithm Of 2*»);
- плюс бесконечность («*Positive Infinity*»);
- постоянная Планка («*Planck Constant (J/Hz)*»);
- элементарный заряд («*Elementary Charge (C)*»);
- скорость света в вакууме («*Speed Of Light In Vacuum (m/sec)*»);
- гравитационная постоянная («*Gravitational Constant (N m<sup>2</sup>/kg<sup>2</sup>)*»);
- постоянная Авогадро («*Avogadro Constant (1/mol)*»);
- Постоянная Ридберга («*Rydberg Constant (/m)*»);
- молярная газовая постоянная («*Molar Gas Constant (J / (mol K))*»).

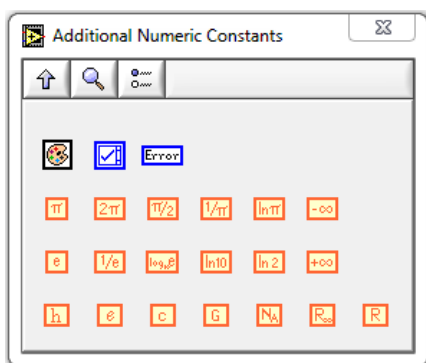


Рисунок 4.2.1.4 – Подраздел дополнительных численных констант («*Additional Numeric Constants*») раздела численных («*Numeric*») функций, размещаемых на блок-диаграммах ВП

## 4.2.2 Использование объединённой арифметики

Объединённая арифметика – функция, относящаяся одновременно к двум разделам: числовому и логическому. Функция объединяет в себе следующие операции: сложение, умножение, логическое сложение (ИЛИ), логическое умножение (И), исключающее ИЛИ.

На Рисунке 4.2.2.1 показано расположение объединённой арифметики в разделе численных функций блок-диаграммы.

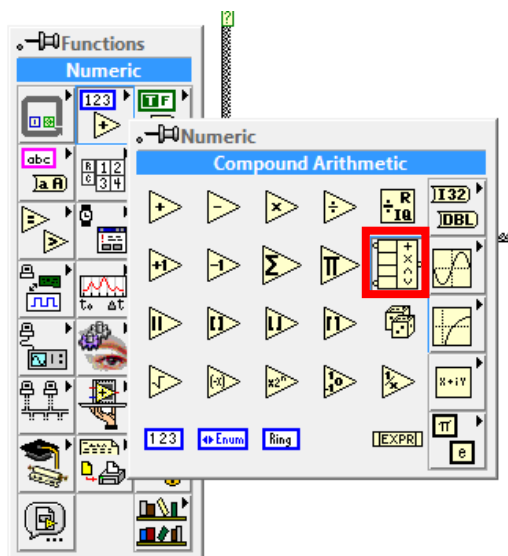


Рисунок 4.2.2.1 – Маркировка расположения функции объединённой арифметики («*Compound Arithmetic*») раздела численных («*Numeric*») функций, размещаемых на блок-диаграмме

При размещении объединённой арифметики на блок-диаграмме по умолчанию доступно только два входа, а также выбрана операция сложения.

Для изменения количества входов необходимо выбрать среди инструментов элемент позиционирования / изменения

размерности / выбора («*Position / Size / Select*») и, ухватившись за нижнюю границу размещённой на блок-диаграмме функции, вытянуть её вниз на нужное количество входов.

Для изменения выполняемой операции функцией объединённой арифметики необходимо вызвать контекстное меню этой функции нажатием на неё правой кнопкой мыши и далее в пункте изменения режима («*Change Mode*») выбрать нужную операцию.

В рамках задания объединённая арифметика используется в режиме логического умножения («*AND*»).

Смена режима объединённой арифметики показана на Рисунке 4.2.2.2.

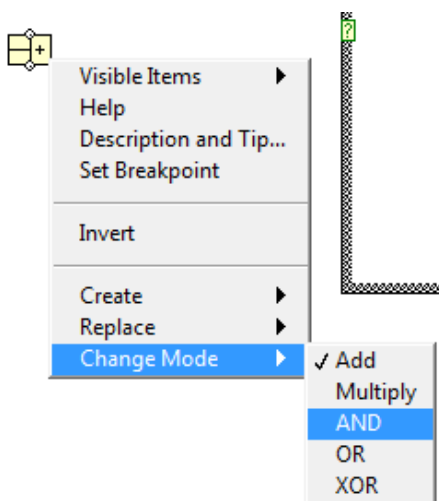


Рисунок 4.2.2.2 – Замена операции в рамках функции объединённой арифметики с численного сложения «+» («*Add*») на логическое умножение «И» («*AND*»)

С целью сокращения количества применяемых логических элементов на блок-диаграмме можно воспользоваться одной из полезных настроек функции объединённой арифметики. В контекстном меню функции объединённой арифметики присутствует инвертирование («*Invert*») полученного значения.

В задании может быть эффективно использована операция отрицания логического умножения («И-НЕ»). Такая операция доступна разработчику ВП, освоившему настройку инвертирования.

На блок-диаграмме размещается объединённая арифметика, работающая в режиме логического «И». В контекстном меню объединённой арифметики устанавливается логический признак инвертирования, как показано на Рисунке 4.2.2.3.

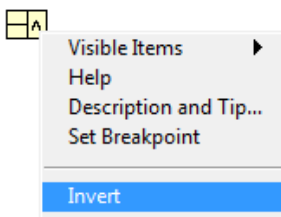


Рисунок 4.2.2.3 – Изменение режима применяемой операции логического умножения с прямой «И» на инверсную «И-НЕ» («*Invert AND*»)

По итогам изучения основ настройки функции объединённой арифметики, а также численных и логических элементов, входящих в состав пакета прикладных программ *National Instruments LabView*, можно переходить непосредственно к созданию заготовки ВП «Умный калькулятор».

#### **4.2.3 Вывод на интерфейсный элемент значений физических постоянных величин, содержащих размерность (единицы измерения)**

Опыт проверки отчётов, составленных обучающимися по результатам выполнения задания «Разработка ВП "Умный калькулятор (*CASE*-калькулятор)"» показал, что разработчики ВП и отчётов по ним сталкиваются с существенными трудностями при реализации ВП по вариантам, содержащим физические константы (нижний ряд формы, изображённой на Рисунке 4.2.1.4).

Во избежание ситуаций, при которых, как правило, обучающиеся запрашивают смену номеров вариантов, в данном параграфе предложен один из возможных способов, позволяющий выполнить вывод значения физической константы на переднюю панель ВП.

Для начала покажем специфику настройки формата и точности («*Format & Precision*») элемента вывода, адаптированного под вывод численных значений физических констант (Рисунок 4.2.3.1).

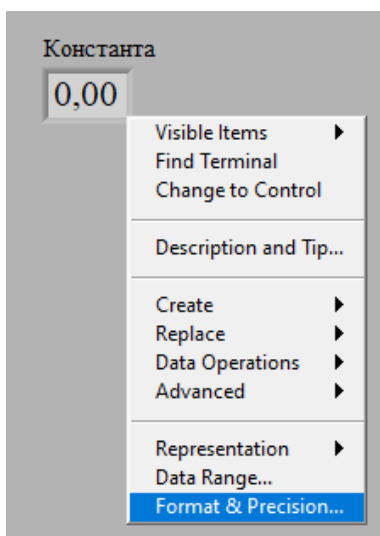


Рисунок 4.2.3.1 – Переход к настройке формата и точности числового элемента вывода

По умолчанию значение точности для отображения вещественного результата в *NI LabView* выставлено до сотых долей (два знака после десятичного разделителя).

Зачастую, физические константы – это либо бесконечно малые, либо бесконечно большие величины. В связи с этим выбор нотации с плавающей точкой («*Floating Point Notation*») – далеко не лучшее решение даже при выставленной точности



вычислений в 15-20 знаков после десятичного разделителя (Рисунок 4.2.3.2).

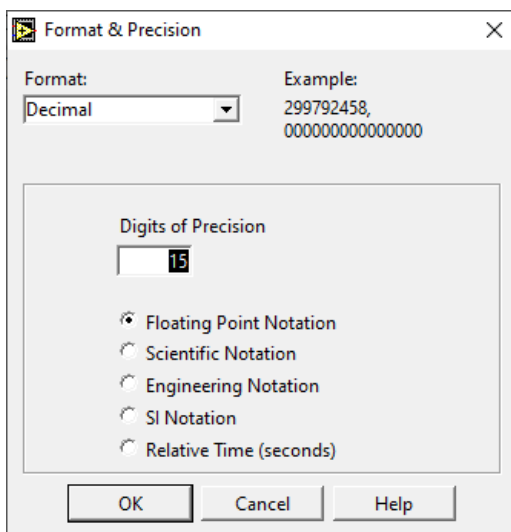


Рисунок 4.2.3.2 – Выставление точности результата на числовом элементе вывода до 15 знаков после десятичного разделителя

Смена настроек, как показано на Рисунке 4.2.3.3, потребует увеличения ширины числового элемента вывода.

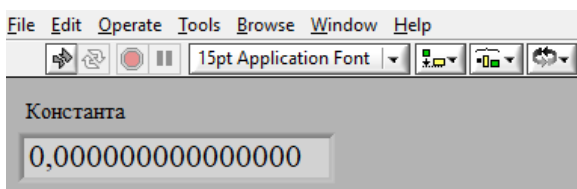


Рисунок 4.2.3.3 – Увеличение ширины числового элемента под вывод значений с точностью до 15 знаков после десятичного разделителя

Перечень дополнительных констант и способ перехода к нему описан в подразделе 4.2.1. На Рисунке 4.2.3.4 продемонстрирован выбор постоянной Планка [22].

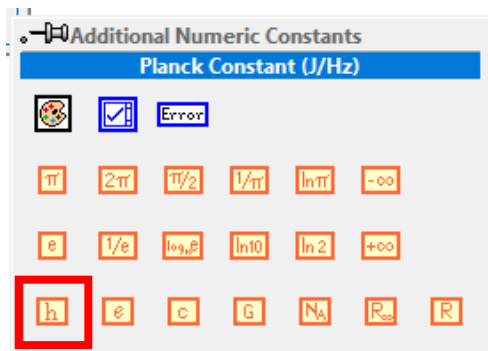


Рисунок 4.2.3.4 – Выбор постоянной Планка из перечня дополнительных числовых констант

При размещении постоянной Планка [22] на блок-диаграмме ВП и попытках подключения её к числовому элементу вывода соединение будет отображаться ошибочным / пунктирным (несовпадение типов), поскольку константа содержит в своём составе размерность (единицы измерения).

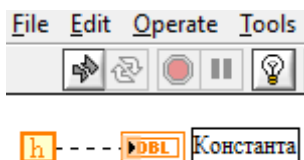


Рисунок 4.2.3.5 – Попытка прямого подключения постоянной Планка к числовому элементу вывода

Принцип работы числового элемента вывода состоит в том, что он в состоянии отображать на передней панели ВП численные значения указанного формата, но в него по умолчанию не заложено отображение единиц измерения.

Одним из известных способов демонстрации значений констант, в которые заложены единицы измерения, является приведение их к безразмерным величинам [23].

Этот способ основывается на математических манипуляциях, изучаемых в школьном курсе физики: для исключения из величины единиц измерения необходимо разделить эту величину на указанные в ней единицы измерения.

Графически закодировать величину интересующей размерности в *NI LabView* не сложно, а разделить на неё исходную константу – ещё более простая задача.

Элемент добавления единиц измерения располагается в разделе числовых функций, в подразделе конвертации. Полный путь к элементу: «*Numeric > Conversion > Convert Unit*» (Рисунок 4.2.3.6).

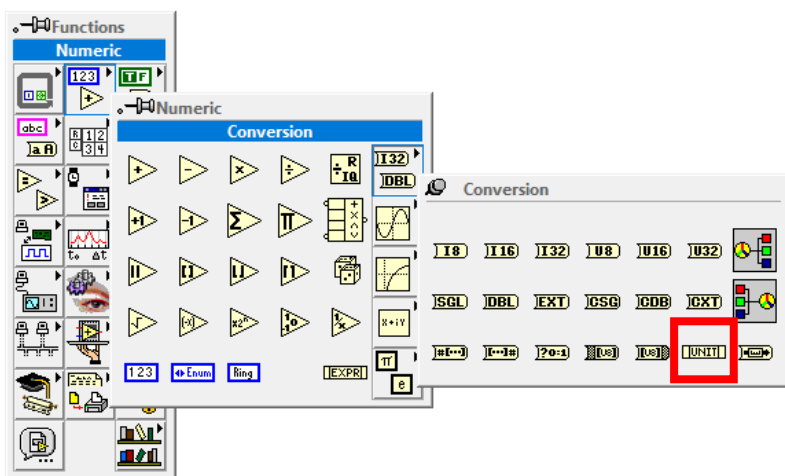


Рисунок 4.2.3.6 – Путь к разделу конвертации с отметкой функции назначения числовому значению единиц измерения

Элемент «*Convert Unit*» последовательно составлен из входной вещественной клеммы, поля для ввода текста (в него и вводится размерность) и выходной вещественной клеммы.

Если внимательно посмотреть на Рисунок 4.2.3.4, то при подсвечивании константы постоянной Планка [22] можно увидеть в скобках закреплённую за ней размерность (« $J/Hz$ », Джоуль делить на Герц, но в русскоязычных учебниках чаще используется «Дж\*с» – Джоуль умножить на секунду). Обучающимся в рамках самостоятельной работы предлагается проверить, равноценны ли эти размерности в пакете прикладных программ *NI LabView*.

На Рисунке 4.2.3.7 показан пример графического кода для приведения постоянной Планка [22] к безразмерной величине [23].

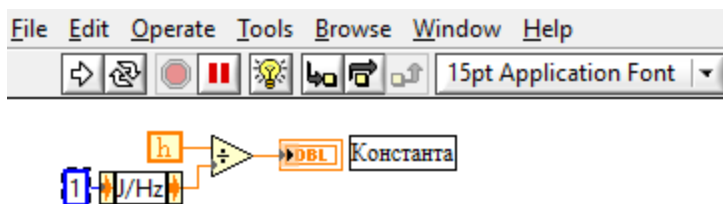


Рисунок 4.2.3.7 – Пример исключения единиц измерения из величины постоянной Планка

Такая программа может быть запущена, однако результат по сравнению с видом, представленным на Рисунке 4.2.3.3, останется неизменным.

Безусловно, указанной точности в 15 знаков после десятичного разделителя для величины постоянной Планка [22] недостаточно. Решить проблему можно повышением точности до большего количества знаков после десятичного разделителя, однако аккуратнее и красивее результат получится, если сменить нотацию с «Плавающей точки» (*Floating Point Notation*) на «Научную» (*Scientific Notation*).

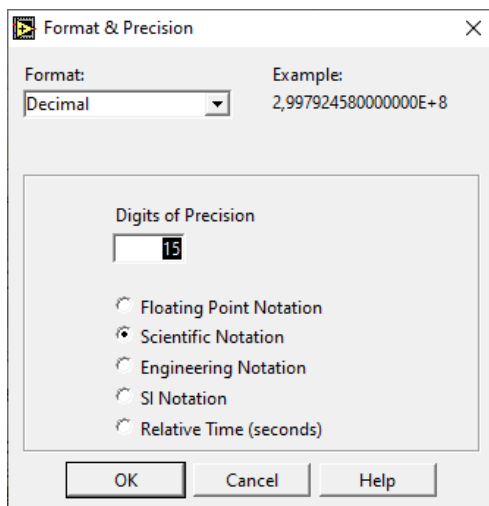


Рисунок 4.2.3.8 – Смена нотации (формата вывода численного результата) на «Научную» (*Scientific*) с сохранением того же количества знаков после десятичного разделителя

Интерфейсные изменения в ВП должны мгновенно сказаться на отображении результата без повторного запуска кода на исполнение.

На переднюю панель будет выведено значение, аналогичное представленному на Рисунке 4.2.3.9. Если округлить его до двух знаков после десятичного разделителя, то получится в точности значение, которое требуют для заучивания авторы большинства учебников школьной физики [24].

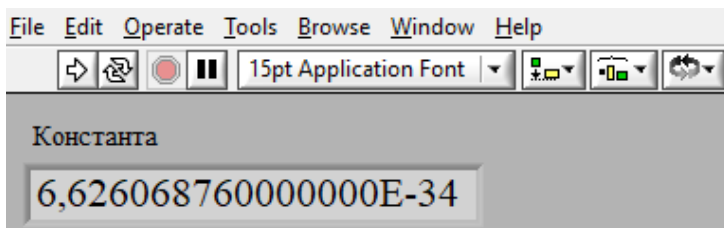


Рисунок 4.2.3.9 – Результат вывода значения постоянной Планка при исключении единиц измерения

Пример постоянной Планка [22] удачен только для демонстрации работы различных нотаций при форматировании вывода результата, однако он не раскрывает всей специфики работы с размерностями в *NI LabView*, поскольку у физической величины постоянной Планка [22] достаточно простая размерность, которая уместается в рамки одного элемента «*Convert Unit*».

Для полноты освещения вопроса работы с размерностями в *NI LabView* рассмотрим пример гравитационной постоянной «*G*» [25] (Рисунок 4.2.3.10).

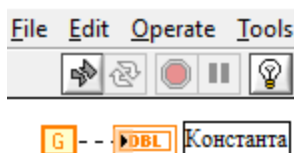


Рисунок 4.2.3.10 – Попытка прямого подключения гравитационной постоянной к числовому элементу вывода

При подсвечивании гравитационной постоянной (Рисунок 4.2.3.11) можно увидеть её размерность ( $N \cdot m^2/kg^2$ ) чуть ниже заглавия экранной формы. Здесь отечественные и зарубежные физики сумели договориться – « $H \cdot m^2 / kg^2$ ».

При вводе такой размерности в один единственный элемент добавления размерности численной величине («*Convert Unit*») сохраняется ошибка связи, аналогичная представленной на Рисунке 4.2.3.10.

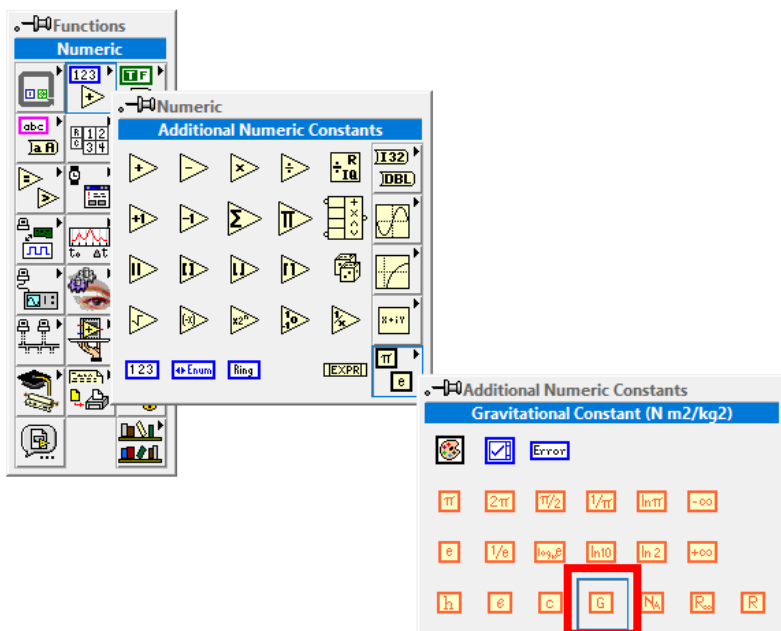


Рисунок 4.2.3.11 – Выбор гравитационной постоянной из перечня дополнительных числовых констант

Проблема ввода сложной размерности в графическом коде, размещаемом на блок-диаграмме ВП, решается путём декомпозиции этой размерности на составляющие с последующим объединением знаками арифметических операций (Рисунок 4.2.3.12).

Представленное на Рисунке 4.2.3.12 демонстрирует работоспособность предложенного решения.

Вероятно, обучающиеся сумеют отыскать и более изящное решение в рамках выполнения самостоятельной работы.

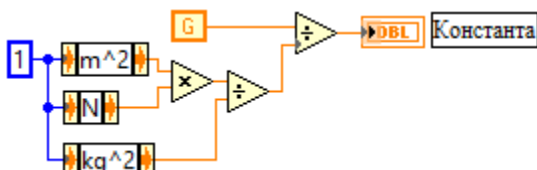


Рисунок 4.2.3.12 – Декомпозиция размерности гравитационной постоянной с последующим её исключением из величины для передачи на числовой элемент вывода

При настройках формата результата, аналогичных представлению, продемонстрированному на Рисунке 4.2.3.8, вывод безразмерной величины [23] на переднюю панель ВП получится аналогичным изображённому на Рисунке 4.2.3.13.

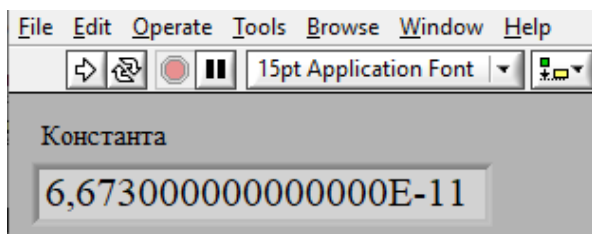


Рисунок 4.2.3.13 – Значение гравитационной постоянной, выведенное в «Научной» нотации на переднюю панель ВП после исключения единиц измерения из физической величины



#### 4.2.4 CASE-структура

Рассмотрение численных и логических элементов уже затрагивалось в заданиях «9» и «10» «Учебной практики».

Структуры в них не использовались, поскольку вычислительный процесс был достаточно простым – линейным. При реализации ВП «Умный калькулятор» требуется разветвление вычислительного процесса, а потому необходимы знания об элементах и структурах, позволяющих организовать разветвление.

Рассмотрим расположение структур *NI LabView* в перечне функций, размещаемых на блок-диаграммах ВП (Рисунок 4.2.4.1).

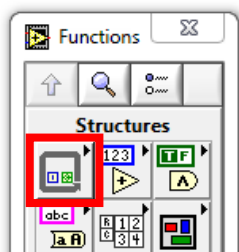


Рисунок 4.2.4.1 – Маркировка раздела структур в перечне функций, размещаемых на блок-диаграммах ВП

Вход в раздел структур сопровождается выводом на экран формы, содержащей все существующие в *NI LabView* структуры (их не много). При чтении раздела слева направо и сверху вниз можно увидеть следующие (Рисунок 4.2.4.2):

- структура последовательности («*Sequence Structure*»);
- CASE-структура («*Case Structure*»);
- структура цикла по известному диапазону значений («*For Loop*»);
- структура цикла с предусловием / постусловием («*While Loop*»);
- узел-формула («*Formula Node*»);

- структура событий («*Event Structure*»);
- глобальная переменная («*Global Variable*»);
- локальная переменная («*Local Variable*»).

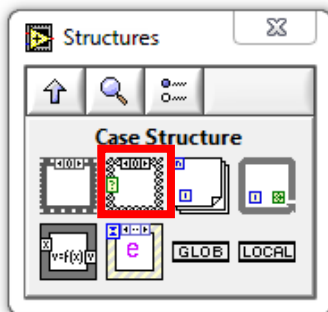


Рисунок 4.2.4.2 – Содержимое раздела структур блок-диаграммы с маркировкой *CASE*-структуры («*Case Structure*»)

*CASE*-структура может быть использована в логическом режиме (к знаку вопроса, расположенному на границе структуры слева, подключается связь логического типа), в таком случае её работа аналогична работе условного оператора («*if*») из структурного программирования.

*CASE*-структура может быть использована в численном режиме (к знаку вопроса, расположенному на границе структуры слева, подключается связь целочисленного типа), и в таком случае структура работает как оператор переключения («*switch*» на языке *Visual C#*).

### 4.3 Пример выполнения задания

#### 4.3.1 Реализация ВП с выводом различных констант на базе *CASE*-структуры

Для демонстрации работы *CASE*-структуры в логическом режиме намеренно откажемся от использования операндов

«Операнд 1» и «Операнд 2», чтобы сделать акцент на логике работы ВП (Рисунок 4.3.1.1).

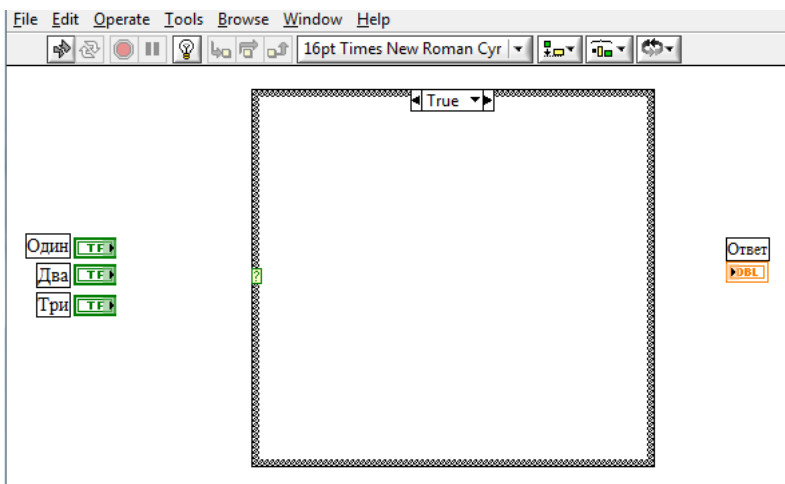


Рисунок 4.3.1.1 – Пример заготовки на блок-диаграмме ВП фрагмента «Умного калькулятора», не требующего участия каких-либо операндов (и операций, соответственно)

Для вывода на числовой элемент вывода конкретного результата оператором должна быть нажата одна и только одна, конкретная кнопка из трёх предложенных на передней панели ВП. Другие кнопки не должны быть нажаты. Должна гарантироваться однозначность (один входной сигнал – один результат), то есть:

- либо кнопка «Один» нажата, кнопки «Два» и «Три» не нажаты,
- либо кнопка «Два» нажата, кнопки «Один» и «Три» не нажаты,
- либо кнопка «Три» нажата, кнопки «Один» и «Два» не нажаты.

Построим схему на блок-диаграмме в соответствии со сформулированной выше логикой. Начнём с первого условия: «Два» и «Три» логически перемножаются и инвертируются функцией объединённой арифметики. Полученный результат логически умножается на «Один» (Рисунок 4.3.1.2).

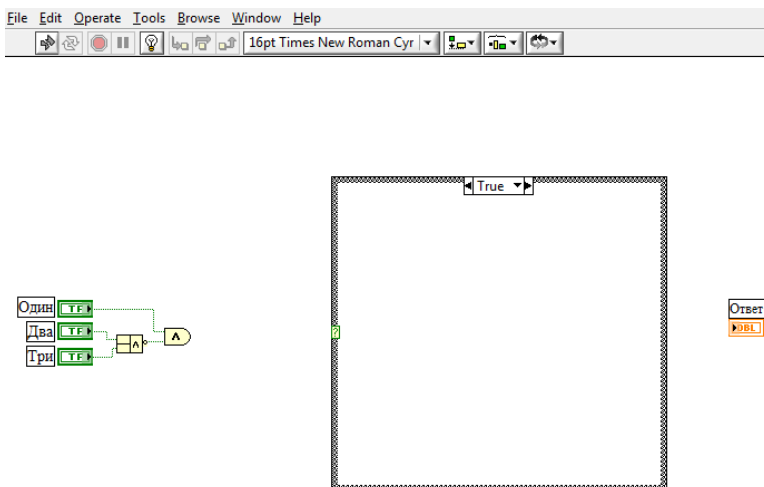


Рисунок 4.3.1.2 – Составление логики для нажатой кнопки «Один» и не нажатых кнопок «Два» и «Три»

По аналогии строится (на блок-диаграмме дополнение размещено ниже) схема для второго условия: «Один» и «Три» логически перемножаются и инвертируются функцией объединённой арифметики. Полученный результат логически умножается на «Два» (Рисунок 4.3.1.3).

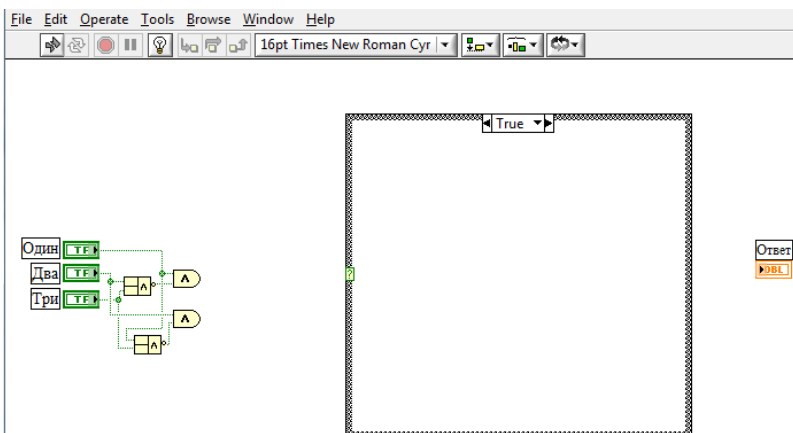


Рисунок 4.3.1.3 – Добавление логики для нажатой кнопки «Два» и не нажатых кнопок «Один» и «Три»

Схема дополняется (на блок-диаграмме дополнение размещено выше) учётом третьего условия: «Один» и «Два» логически перемножаются и инвертируются функцией объединённой арифметики. Полученный результат логически умножается на «Три» (Рисунок 4.3.1.4).

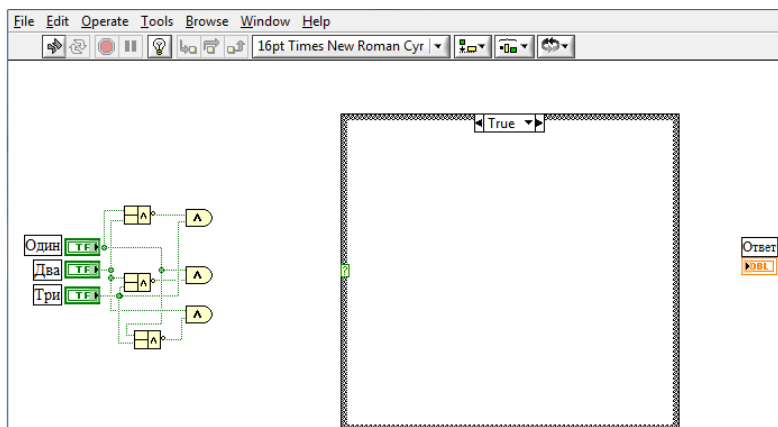


Рисунок 4.3.1.4 – Добавление логики для нажатой кнопки «Три» и не нажатых кнопок «Один» и «Два»

Решение задачи контроля нажатия одной единственной кнопки из трёх при составлении данной логической цепочки – это тот редкий случай, когда уместно применение операции исключающего «ИЛИ».

Это трудно принять с первого прочтения, поскольку само по себе в рассуждениях напрашивается обыкновенное логическое сложение, но это не так.

Для доказательства составим таблицу истинности по логическому сложению (Таблица 4.3.1.1) и таблицу истинности для исключающего «ИЛИ» (Таблица 4.3.1.2).

Таблица 4.3.1.1 – Истинность объединения трёх операндов по «ИЛИ»

№ стр.	Один	Два	Три	«OR»
1	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>
2	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
3	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
4	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>
5	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
6	<i>true</i>	<i>false</i>	<i>true</i>	<i>true</i>
7	<i>true</i>	<i>true</i>	<i>false</i>	<i>true</i>
8	<i>true</i>	<i>true</i>	<i>true</i>	<i>true</i>

Таблица 4.3.1.2 – Истинность объединения трёх операндов по исключающему «ИЛИ»

№ стр.	Один	Два	Три	XOR
1	<i>false</i>	<i>false</i>	<i>false</i>	<i>false</i>
2	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>
3	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
4	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>
5	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
6	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
7	<i>true</i>	<i>true</i>	<i>false</i>	<i>false</i>
8	<i>true</i>	<i>true</i>	<i>true</i>	<i>false</i>

Видно, что объединение по «ИЛИ» – это истинность хотя бы одного из трёх. То есть, если хотя бы один элемент отвечает истинной, то всё логическое выражение отвечает истинной – логика не подходит. Исключающее «ИЛИ» гласит: только один должен ответить истинной, тогда всё выражение ответит истинной.

Сцепление ранее сформулированных условий по исключающему «ИЛИ», настроенному в элементе объединённой арифметики, представлено на Рисунке 4.3.1.5. Вместе с тем для тестирования ВП в блок истинного результата всего выражения устанавливается константа «плюс бесконечность».

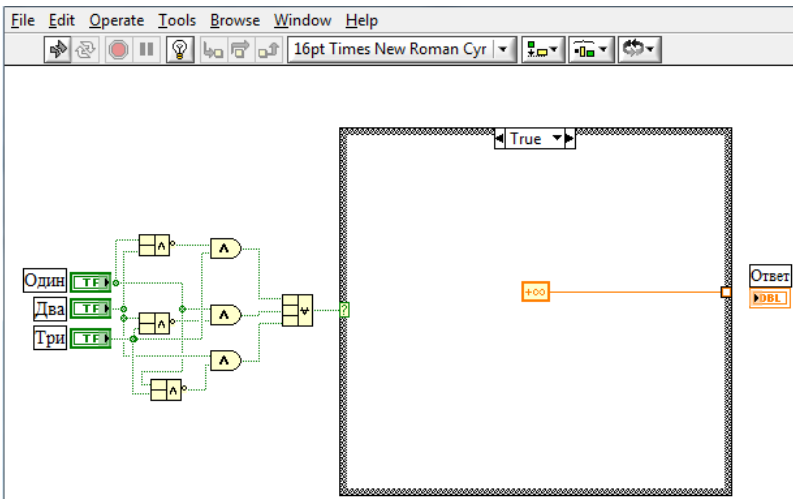


Рисунок 4.3.1.5 – Объединение условий по исключающему «ИЛИ» («XOR»).

В блоке ложного результата (Рисунок 4.3.1.6) всего выражения выставляется константа не числового значения («NaN»).

Для создания константы не числового значения на блок-диаграмме размещается обыкновенная целочисленная константа и в ней вручную записывается с клавиатуры «NaN» при строгом

учёте верхнего и нижнего регистра символов (две заглавных «N» объединяются между собой строчной «a»).

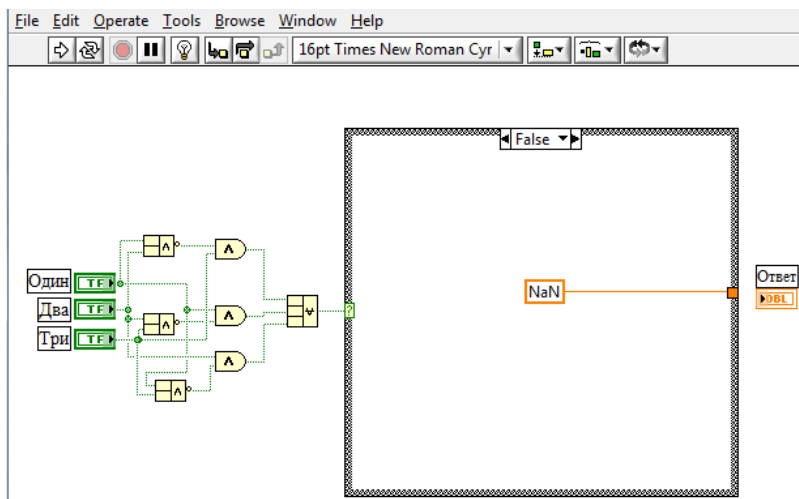


Рисунок 4.3.1.6 – Проверка работоспособности составленной схемы: если условие не выполняется – выводится «не числовое значение» («NaN»)

После завершения размещения содержимого во все блоки CASE-структуры не забудьте в каждом соединить это содержимое с объединённым выходом CASE-структуры, который и должен сопрягаться с численным элементом вывода «Ответ».

Объединённый выход CASE-структуры на Рисунке 4.3.1.6 отмечен оранжевым квадратом на её правой границе, напротив элемента вывода «Ответ». На Рисунке 4.3.1.7 соответствующее сопряжение элемента вывода «Ответ» с объединённым выходом CASE-структуры выполнено.

Отсутствие соединения содержимого одного или нескольких блоков с объединённым выходом CASE-структуры – это частая ошибка обучающихся, которая препятствует запуску ВП на исполнение. Заметить её можно, обратив внимание на



«выколотый» объединённый выход, каким его можно видеть на Рисунке 4.3.1.5.

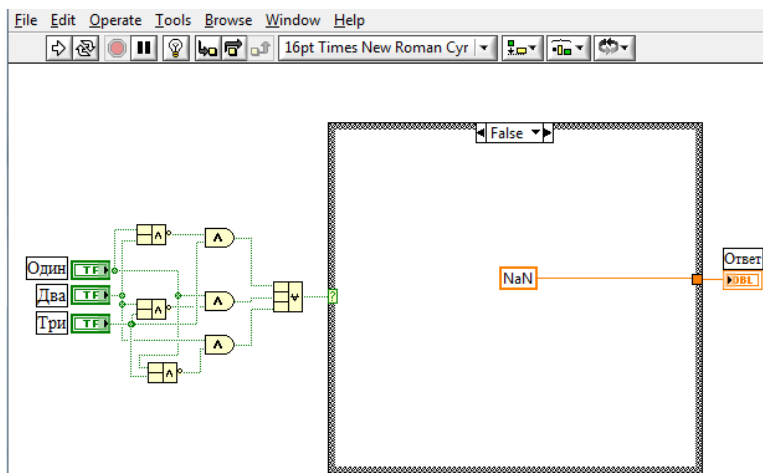


Рисунок 4.3.1.7 – Передача сформированного CASE-структурой результата на элемент вывода «Ответ»

Запуск ВП при всех не нажатых кнопках даёт в результате не числовое значение – произошло попадание в блок ложного результата (*False*). Согласно составленной ранее таблице истинности (Таблица 4.3.1.2) так и должно быть (Рисунок 4.3.1.8).

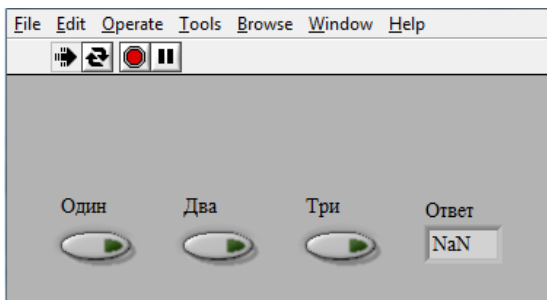


Рисунок 4.3.1.8 – Проверка работоспособности составленной логики: ничего не нажато – результат неопределённый («NaN»)

Запуск ВП при всех нажатых кнопках тоже даёт в результате не числовое значение – попадание в блок ложного результата (*False*) CASE-структуры (Рисунок 4.3.1.9).

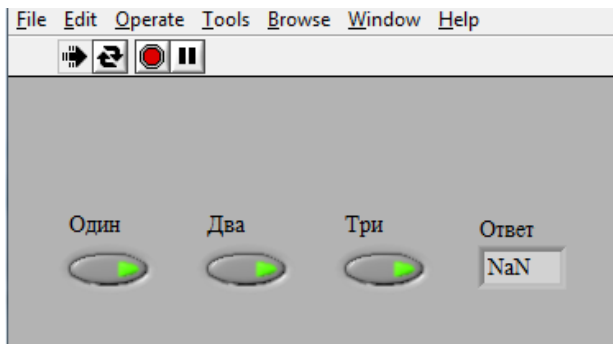


Рисунок 4.3.1.9 – Проверка работоспособности составленной логики: нажаты все кнопки сразу – результат неопределённый («NaN»)

Запуск ВП с нажатыми кнопками «Один» и «Три» – шестая строчка таблицы истинности (не считая заголовка). Логический ответ – «ложь» (Рисунок 4.3.1.10).

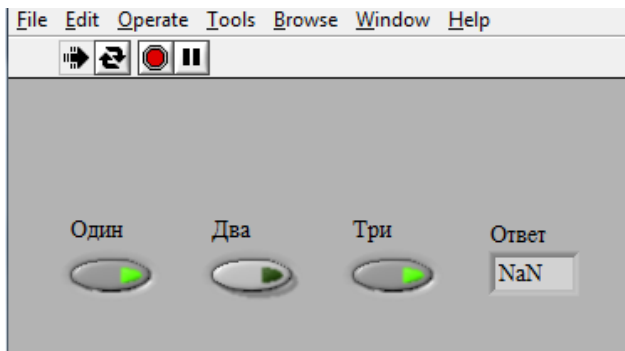


Рисунок 4.3.1.10 – Проверка работоспособности составленной логики: нажаты сразу две кнопки – результат неопределённый («NaN»)

Запуск ВП с нажатыми «Один» и «Два» – седьмая строка таблицы истинности (не считая заголовка). В ответе – «ложь». Адекватно (Рисунок 4.3.1.11).

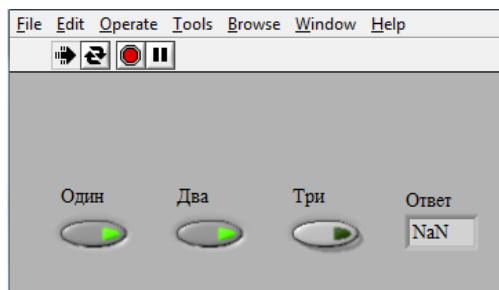


Рисунок 4.3.1.11 – Проверка работоспособности составленной логики: нажаты сразу две кнопки – результат неопределённый («NaN»)

Нажатие только кнопки «Один» – это пятая строка таблицы истинности. В соответствующем блоке (*True*) размещена константа «плюс бесконечность». Вместо не числового значения, как это было показано при проверке всех тестовых примеров, запущенных ранее, на элементе вывода можно увидеть надпись «*Inf*» (сокращение от «*Infinity*»), закреплённую за этой константой. Адекватно (Рисунок 4.3.1.12).

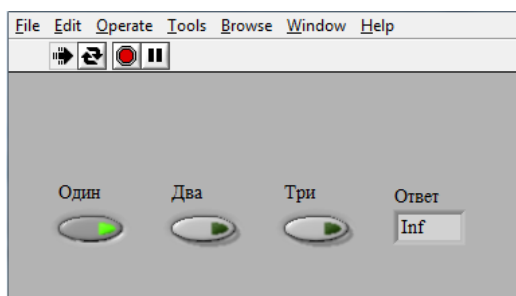


Рисунок 4.3.1.12 – Проверка работоспособности составленной логики: нажата только одна кнопка – результат определён («плюс бесконечность»)

После успешного тестирования реакции на однозначность входного сигнала продолжим развитие кода заготовки ВП «Умный калькулятор» в направлении большей конкретики.

В блоке ложного результата (*False*) контрольной (внешней) CASE-структуры остаётся константа не числового значения («NaN»), а в блоке истинного результата (*True*) размещаем ещё одну, вложенную CASE-структуру.

На логический вход вложенной CASE-структуры подаём ветвь результата логики, отвечающей за нажатие кнопки «Один» и не нажатие кнопок «Два» и «Три».

Итого, если нажата кнопка «Один», в блоке истинного результата должна зарождаться численная константа, равная «1» (Рисунок 4.3.1.13).

Если «Один» не нажата, то остаётся ещё два возможных варианта, которые необходимо проверить (либо нажата «Два», либо нажата «Три»). Это ещё одна вложенная CASE-структура.

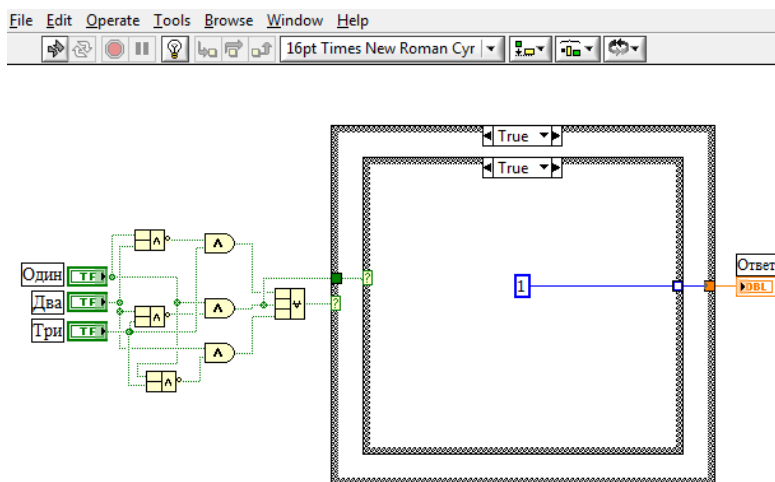


Рисунок 4.3.1.13 – Развитие структуры программы: в случае, если нажата только одна кнопка, и эта кнопка – «Один». Должно выводиться целочисленное значение, равное «1»

На логический вход новой *CASE*-структуры подаём проверку условия нажатия на «Два» (Рисунок 4.3.1.14). Если, действительно, нажата «Два», то должна генерироваться константа, равная «2».

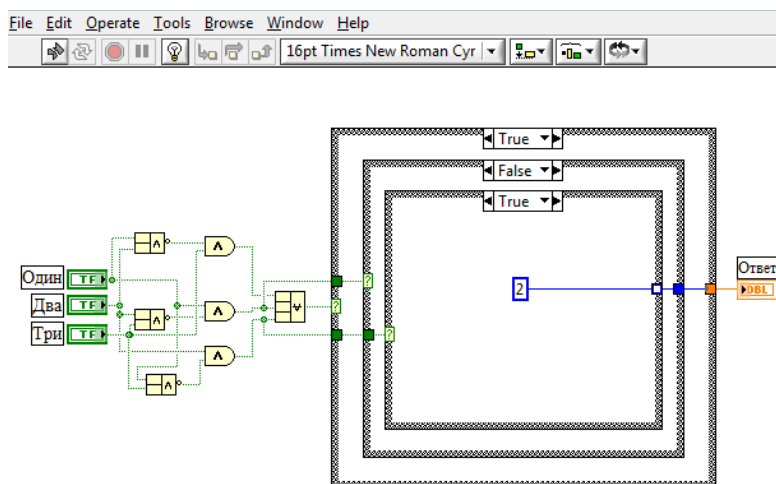


Рисунок 4.3.1.14 – Развитие структуры программы: в случае, если нажата только одна кнопка, и эта кнопка не «Один», а «Два», то должно выводиться целочисленное значение – «2»

Но если не «Два», то остаётся один единственный вариант, что нажата «Три». Дополнительной *CASE*-структурой его контролировать не нужно. Константу со значением «3» генерирует оставшийся незаполненным блок ложного результата (Рисунок 4.3.1.15).

У неопытных программистов может возникнуть желание надстроить ещё одну *CASE*-структуру с целью проверки условия для ветки «Три» по аналогии. Это не будет ошибкой, но останется избыточный блок ложного результата, в который никогда и ни при каких обстоятельствах вычислительный процесс ВП не попадёт.

Вообще говоря, с избыточностью в коде согласно правилам вежливости программиста (Приложение 2) необходимо бороться.

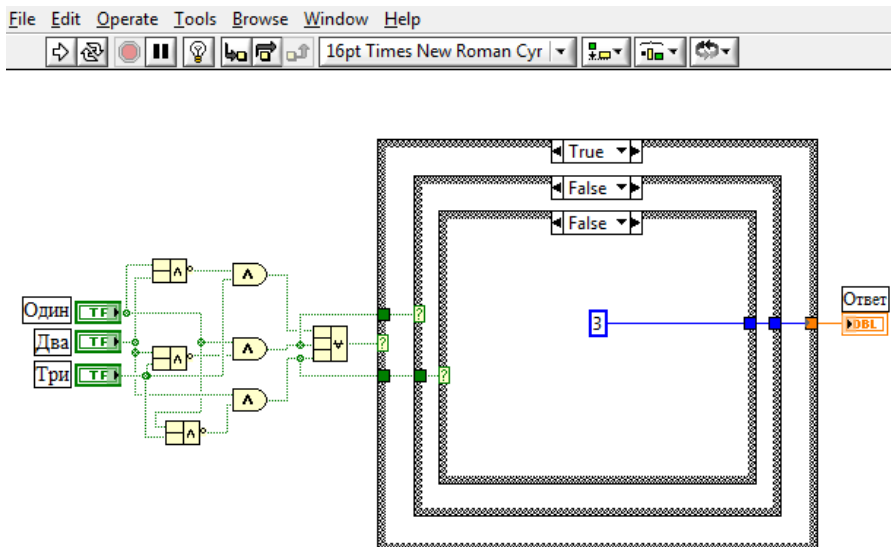


Рисунок 4.3.1.15 – Развитие структуры программы: в случае, если нажата только одна кнопка, и эта кнопка не «Один» и не «Два», то остаётся заключительный вариант – нажата кнопка «Три». Должно выводиться в качестве ответа целочисленное значение, равное «3»

Нижеследующие иллюстрации (Рисунки 4.3.1.16-4.3.1.18) демонстрируют адекватность работы созданного фрагмента ВП «Умный калькулятор» с учётом ранее проведенных тестов по защите от одновременного выбора пользователем нескольких операций.

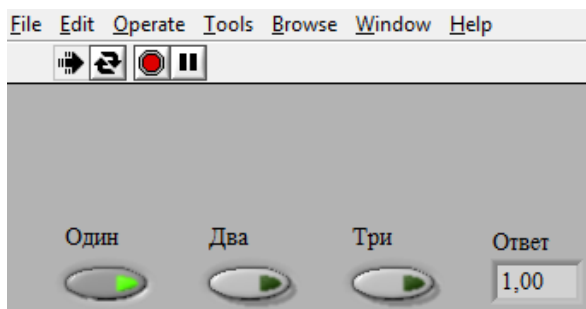


Рисунок 4.3.1.16 – Проверка работоспособности программного обеспечения: нажата кнопка «Один» и в качестве ответа выдаётся значение, равное «1»

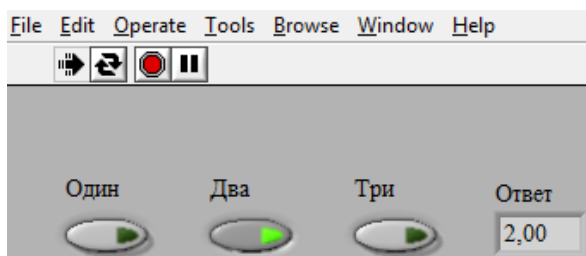


Рисунок 4.3.1.17 – Проверка работоспособности программного обеспечения: нажата кнопка «Два» и в качестве ответа выдаётся значение, равное «2»

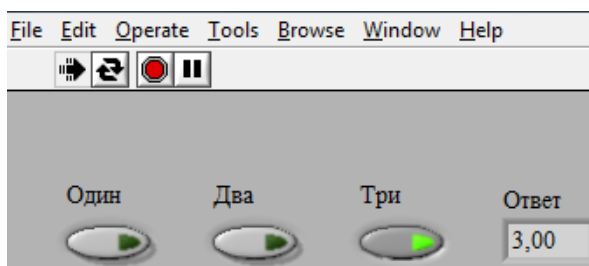


Рисунок 4.3.1.18 – Проверка работоспособности программного обеспечения: нажата кнопка «Три» и в качестве ответа выдаётся значение, равное «3»

### 4.3.2 Реализация ВП на базе оператора выбора с двумя операндами и тремя операциями

Задание номер «11» ориентировано на приобретение обучающимися навыков работы со структурой *CASE*, однако это не означает, что она (структура) является единственным способом решения задачи разветвления вычислительного процесса.

Автор продолжает придерживаться мнения, что структура *CASE* является более читаемой и более гибкой для всех случаев, когда возникает необходимость развития и без того уже разветвлённой логики – добавления новых условий в контур управления.

Менее гибкой, менее читаемой, но более компактной компонентой пакета прикладных программ *NI LabView*, позволяющей реализовать разветвление вычислительного процесса является функция выбора «*Select*» (Рисунок 4.3.2.1).

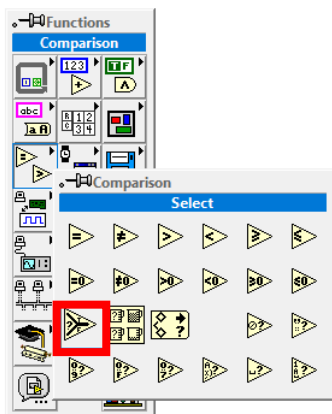


Рисунок 4.3.2.1 – Маркировка функции выбора («*Select*») в разделе функций сравнения, размещаемых на блок-диаграмме ВП

Данный параграф призван помочь обучающимся, которым решение задачи при помощи функции выбора («*Select*»)



покажется более простым по сравнению с применением *CASE*-структуры.

В первую очередь необходимо ввести проверку нажатия хотя бы одной из кнопок операций, выполняемых над парой операндов. Если ни одна из операций не выполняется, то на элемент вывода направляется сигнал не числового значения «NaN» (он необходим для понимания, что код, действительно, был запущен на исполнение, но ни одна из операций над операндами при этом не была выполнена). Этот шаг на Рисунке 4.3.2.2 объединён с проверкой и обработкой операции сложения двух операндов.

Логика следующая: «Если нажата хотя бы одна из кнопок операций, то интересно знать, какая именно она, – эта операция? Сложение? – Действительно, сложение, – тогда его и выполняем. Не сложение – тогда что же это?»

На данном этапе мы не знаем и знать не можем, что делать во всех остальных ситуациях, которые ещё не рассмотрены, но являются однозначными (выбрано вычитание или выбрано умножение). При этом проверить готовность составленного алгоритма к дальнейшему развитию уже можно.

В условиях отсутствия ветвей дальнейшего принятия решения для вычислительного процесса на этом месте графического кода выставлена «заглушка» (временный «терминатор») в виде константы «плюс бесконечность».

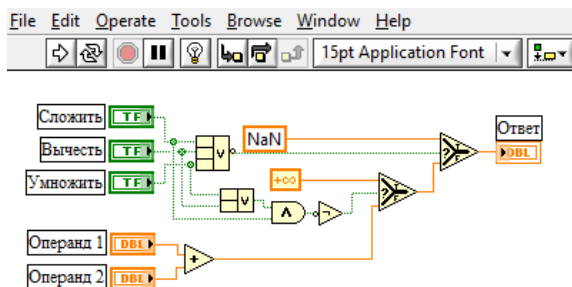


Рисунок 4.3.2.2 – Реализация первого звена контроля нажатия кнопок, а также реакции на нажатие кнопки выбора операции сложения

На Рисунке 4.3.2.3 показано, как структура, представленная на Рисунке 4.3.2.2 может быть развита для детектирования выбора операции вычитания (нажата кнопка операции «Вычесть» и не нажаты кнопки операций «Сложить» или «Умножить»).

Такой код ВП может быть запущен на исполнение, но он не будет обрабатывать операцию умножения.

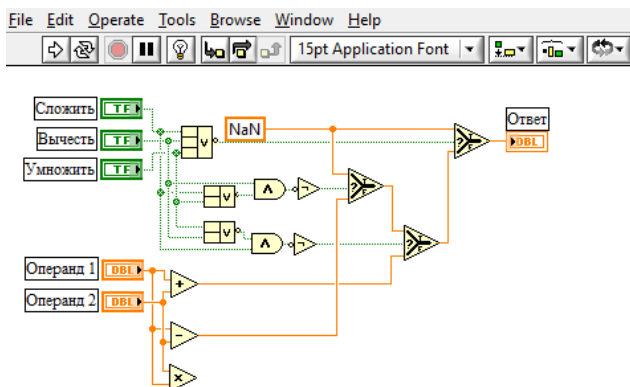


Рисунок 4.3.2.3 – Развитие схемы – реализация реакции на нажатие кнопки выбора операции вычитания (при нажатии на несколько кнопок – генерируется сигнал ошибки)

Отмеченная выше проблема отсутствия детектирования операции умножения решена на Рисунке 4.3.2.4 дополнением кода ещё одним звеном логики, в котором проверяется факт того, что нажата кнопка операции «Умножить» и не нажаты кнопки операций «Сложить» или «Вычесть».

Развитие структуры одновременно решает и задачу выбора пользователем нескольких операций, когда нажатыми оказываются несколько кнопок. Сигнал на выходе при этих обстоятельствах аналогичен тому, который направляется на выход при отсутствии выбора какой-либо из предусмотренных операций.

Согласно формулировке задания (раздел 4) на эту позицию обучающимся предлагается разместить целочисленный код ошибки.

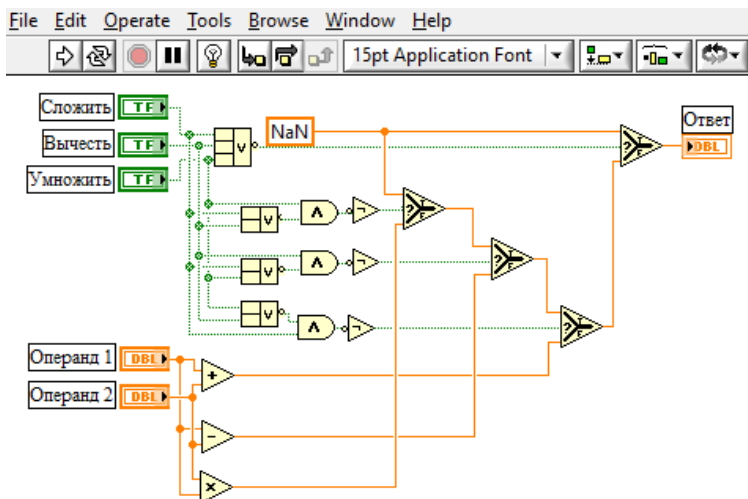


Рисунок 4.3.2.4 – Завершение схемы – отлажен корректный отклик на все возможные и интересные ситуации

Обучающимся не следует забывать о подготовке тестовых примеров и о фактическом тестировании составленного программного обеспечения на их основе (Приложение 2).

На Рисунках 4.3.2.5-4.3.2.8 проведена только часть процедуры тестирования ВП. Внимание уделено только ситуациям, когда: не выбрана ни одна из операций, выбрано только сложение, выбрано только вычитание, выбрано только умножение.

Без внимания остались ситуации, когда одновременно выбраны все три операции, когда одновременно нажаты кнопки операций сложения и вычитания, сложения и умножения, вычитания и умножения. Всего восемь возможных ситуаций ( $2^3$ ) для тестирования.

Обучающимся рекомендуется проверить все восемь в рамках выполнения самостоятельной работы, а при решении индивидуальной части задания в обязательном порядке рассмотреть все возможные ситуации при тестировании составленного программного обеспечения.

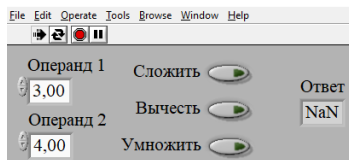


Рисунок 4.3.2.5 – Вывод результата («NaN») при отсутствии выбора операции (не нажата ни одна из кнопок)

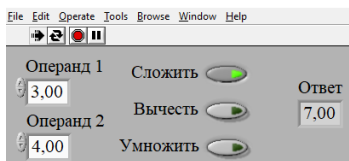


Рисунок 4.3.2.6 – Вывод результата сложения значений двух операндов (выбрана операция «Сложить»)

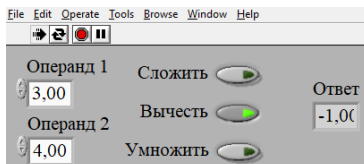


Рисунок 4.3.2.7 – Вывод результата вычитания значений двух операндов (выбрана операция «Вычитать»)

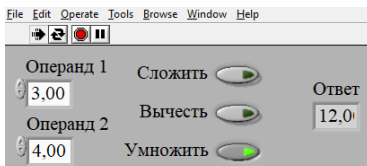


Рисунок 4.3.2.8 – Вывод результата умножения значений двух операндов (выбрана операция «Умножить»)

#### 4.4 Варианты индивидуального задания

Таблица 4.4.1 – Исходные данные для выполнения индивидуального задания

№ вар.	Содержание работы
1.	Использовать <b>три</b> операнда. Предусмотреть операции:  – сложения; – инверсии суммы; – инверсии суммы объединённой арифметикой; – сложения объединённой арифметикой.
2.	Использовать <b>два</b> операнда. Предусмотреть операции:  – умножения; – деления первого на второй; – деления второго на первый; – масштабирования на степень двойки; – логарифма второго по основанию первого.
3.	Использовать <b>один</b> операнд. Предусмотреть операции:  – абсолютного значения; – сигнума; – синуса; – экспонирования.
4.	Использовать <b>два</b> операнда. Предусмотреть операции:  – деления по модулю первого; – деления по модулю второго; – инкрементирования первого.

№ вар.	Содержание работы
5.	<p>Использовать <b>один</b> операнд.  Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– секанса;</li> <li>– натурального логарифма;</li> <li>– декрементирования;</li> <li>– инверсии;</li> <li>– получения Пифагоровой константы («ПИ»).</li> </ul>
6.	<p>Использовать <b>четыре</b> операнда.  Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– инверсии суммы объединённой арифметикой;</li> <li>– инверсии разности объединённой арифметикой.</li> </ul>
7.	<p>Использовать <b>два</b> операнда.  Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– арктангенса с двумя параметрами;</li> <li>– арктангенса первого;</li> <li>– арктангенса второго;</li> <li>– тангенса первого;</li> <li>– тангенса второго.</li> </ul>
8.	<p>Использовать <b>один</b> операнд.  Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– экспонирования;</li> <li>– взятия логарифма по основанию 2;</li> <li>– натурального логарифма;</li> <li>– десятичного логарифма.</li> </ul>

№ вар.	Содержание работы
9.	<p>Предусмотреть вывод констант:</p> <ul style="list-style-type: none"> <li>– Пифагоровой;</li> <li>– основания натурального логарифма;</li> <li>– гравитационной;</li> <li>– не числового значения;</li> <li>– плюс бесконечности;</li> <li>– минус бесконечности.</li> </ul>
10.	<p>Использовать <b>два</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– остатка от деления первого на второй;</li> <li>– остатка от деления второго на первый.</li> </ul>
11.	<p>Использовать <b>три</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– квадратного корня из суммы квадратов;</li> <li>– квадратного корня из модуля разности квадратов;</li> <li>– квадратного корня из модуля разности;</li> <li>– квадратного корня из модуля суммы.</li> </ul>
12.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– арккосинуса;</li> <li>– округления до ближайшего целого;</li> <li>– квадратного корня.</li> </ul>
13.	<p>Использовать <b>три</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– нахождения периметра треугольника;</li> <li>– нахождения площади треугольника;</li> <li>– нахождения высоты треугольника.</li> </ul>

№ вар.	Содержание работы
14.	<p>Использовать <b>два</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– обращение суммы;</li> <li>– корень квадратный из суммы;</li> <li>– деление второго на первый;</li> <li>– масштабирование на степень двойки;</li> <li>– логарифм второго по основанию первого.</li> </ul>
15.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– нахождения диаметра окружности;</li> <li>– нахождения площади круга;</li> <li>– нахождения длины окружности.</li> </ul>
16.	<p>Использовать <b>четыре</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– нахождения периметра прямоугольника;</li> <li>– нахождения площади прямоугольника;</li> <li>– нахождения диагонали прямоугольника.</li> </ul>
17.	<p>Использовать <b>три</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– нахождения объёма фигуры;</li> <li>– нахождения площади поперечного сечения фигуры;</li> <li>– нахождения площади продольного сечения фигуры.</li> </ul>
18.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– взятия квадратного корня;</li> <li>– тангенса;</li> <li>– обращения;</li> <li>– возведения двойки в степень.</li> </ul>



№ вар.	Содержание работы
19.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– математического округления;</li> <li>– округления в большую сторону;</li> <li>– округления в меньшую сторону.</li> </ul>
20.	<p>Использовать <b>два</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– вывода численной константы, равной «8»;</li> <li>– вычитания второго из первого;</li> <li>– вычитания первого из второго;</li> <li>– суммы синуса первого и косинуса второго (синус и косинус в одной операции).</li> </ul>
21.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– вычисления значения функции отсчётов;</li> <li>– декрементирования.</li> </ul>
22.	<p>Использовать <b>три</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– произведения трёх;</li> <li>– разности суммы первого и третьего из второго;</li> <li>– суммы трёх;</li> <li>– частного произведения второго на третье на первое.</li> </ul>
23.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– косинуса;</li> <li>– степени десяти;</li> <li>– вывода постоянной Планка;</li> <li>– инкрементирования;</li> <li>– декрементирования.</li> </ul>

№ вар.	Содержание работы
24.	<p>Предусмотреть вывод констант:</p> <ul style="list-style-type: none"> <li>– нечисловое значение;</li> <li>– Авогадро;</li> <li>– обращённой Пифагорову;</li> <li>– натурального логарифма десяти.</li> </ul>
25.	<p>Использовать <b>три</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– вычисления периметра треугольника;</li> <li>– вычисления высоты треугольника;</li> <li>– вычисления площади треугольника.</li> </ul>
26.	<p>Использовать <b>один</b> операнд. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– возведения двойки в степень;</li> <li>– котангенса;</li> <li>– вывода обращённого основания натурального логарифма.</li> </ul>
27.	<p>Использовать <b>два</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– синуса суммы;</li> <li>– экспонирования суммы;</li> <li>– обращения суммы;</li> <li>– возведения двойки в степени суммы.</li> </ul>
28.	<p>Использовать <b>шесть</b> операндов. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– произведения объединённой арифметикой;</li> <li>– суммирования объединённой арифметикой.</li> </ul>

№ вар.	Содержание работы
<b>29.</b>	<p>Использовать <b>один</b> операнд.  Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– косеканса;</li> <li>– натурального логарифма, проходящего через начало координат.</li> </ul>
<b>30.</b>	<p>Использовать <b>два</b> операнда.  Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– косинуса разности;</li> <li>– вычитания из первого квадрата тангенса второго.</li> </ul>
<b>31.</b>	<p>Предусмотреть вывод:</p> <ul style="list-style-type: none"> <li>– элементарного заряда;</li> <li>– скорости света в вакууме;</li> <li>– Постоянной Ридберга;</li> <li>– молярной газовой постоянной.</li> </ul>

№ вар.	Содержание работы
32.	<p>Предусмотреть вывод:</p> <ul style="list-style-type: none"> <li>– Пифагоровой константы, умноженной на 2 (одной константой <i>NI LabView</i>);</li> <li>– Пифагоровой константы, умноженной на 2 (сочетанием констант <i>NI LabView</i>);</li> <li>– Пифагоровой константы, делённой на 2 (одной константой <i>NI LabView</i>);</li> <li>– Пифагоровой константы, делённой на 2 (сочетанием констант <i>NI LabView</i>);</li> <li>– натурального логарифма двойки (одной константой <i>NI LabView</i>);</li> <li>– натурального логарифма двойки (сочетанием компонентов <i>NI LabView</i>);</li> <li>– натурального логарифма Пифагоровой константы (одной константой <i>NI LabView</i>);</li> <li>– натурального логарифма Пифагоровой константы (сочетанием компонентов <i>NI LabView</i>).</li> </ul>
33.	<p>Предусмотреть вывод:</p> <ul style="list-style-type: none"> <li>– псевдослучайной величины, округлённой до целого в большую сторону;</li> <li>– псевдослучайной величины, округлённой до десятых;</li> <li>– псевдослучайной величины, округлённой до сотых;</li> <li>– псевдослучайной величины, округлённой до тысячных.</li> </ul>

№ вар.	Содержание работы
34.	<p>Использовать <b>два</b> операнда. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– вычисления суммы двух функций отсчётов, взятых от различных аргументов;</li> <li>– произведения инкремента первого и декремента второго.</li> </ul>
35.	<p>Использовать <b>шесть</b> операндов. Предусмотреть операции:</p> <ul style="list-style-type: none"> <li>– логического «И» объединённой арифметикой;</li> <li>– логического «ИЛИ» объединённой арифметикой.</li> </ul>

#### 4.5 Пример компоновки графического пользовательского интерфейса ВП

В качестве первого приближения к созданию ВП «Умный калькулятор» рассмотрим один из его возможных графических пользовательских интерфейсов.

Для понимания принципов наращивания (масштабирования) структуры ВП «Умный калькулятор» достаточно рассмотреть три операции.

Каждой операции ставится в соответствие логический элемент ввода – кнопка. Так на передней панели должны быть размещены три кнопки, которые назовём слева направо «Один», «Два» и «Три», соответственно. В таком случае подразумевается вывод на численный элемент вывода «Ответ» по нажатию своей, заранее определённой целочисленной константы: либо «1», либо «2», либо «3».

Стоит отметить, что для большинства индивидуальных вариантов данной работы требуется использование операндов для выполнения над ними определённых операций. Под каждый операнд выделяется свой собственный числовой элемент ввода.

Заготовка интерфейса на передней панели ВП, представленная на Рисунке 4.5.1, содержит все вышеуказанные элементы.

В примере, рассмотренном в подразделе 4.3.1, «Операнд 1» и «Операнд 2» не используются.

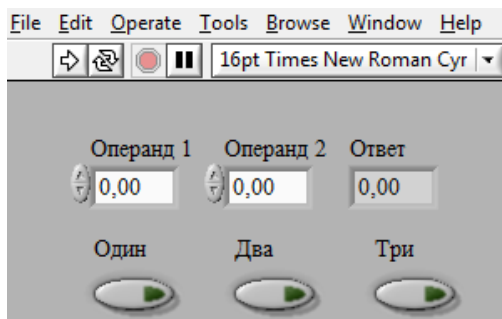


Рисунок 4.5.1 – Пример графического пользовательского интерфейса ВП «Умный калькулятор» с двумя входными операндами, тремя операциями и одним элементом вывода с численным результатом

#### 4.6 Рекомендации по размещению информации на блок-диаграмме ВП

При шаблонном наборе интерфейсных элементов управления, представленных на Рисунке 4.5.1, блок-диаграмма ВП будет содержать их отображения (на Рисунке 4.6.1 для удобства они размещены слева и справа от *CASE*-структуры как входная и выходная информация, соответственно). Они добавляются автоматически. *CASE*-структура должна быть размещена разработчиком вручную.

Заметим, что для большей компактности размещения элементов на блок-диаграмме при использовании инструмента позиционирования / изменения размера / выбора («*Position / Size / Select*») наименования этих элементов размещены слева от них.

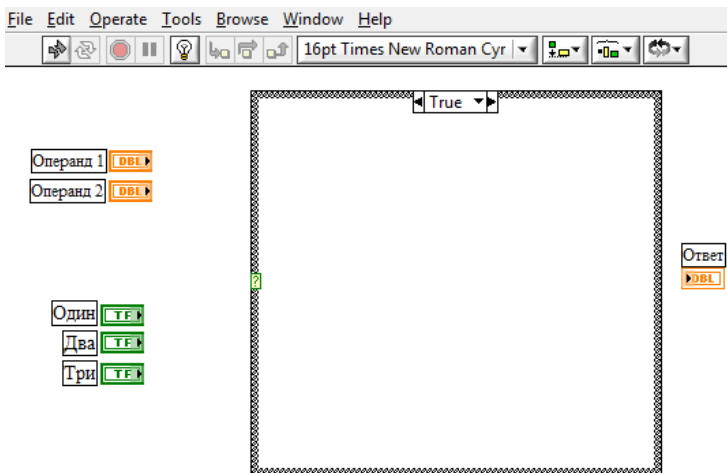


Рисунок 4.6.1 – Заготовка блок-диаграммы ВП

Итого в заготовке использованы элементы из следующих функциональных разделов:

- численные функции;
- логические функции;
- структуры.

4.7 Блок-схемы алгоритма для типового решения

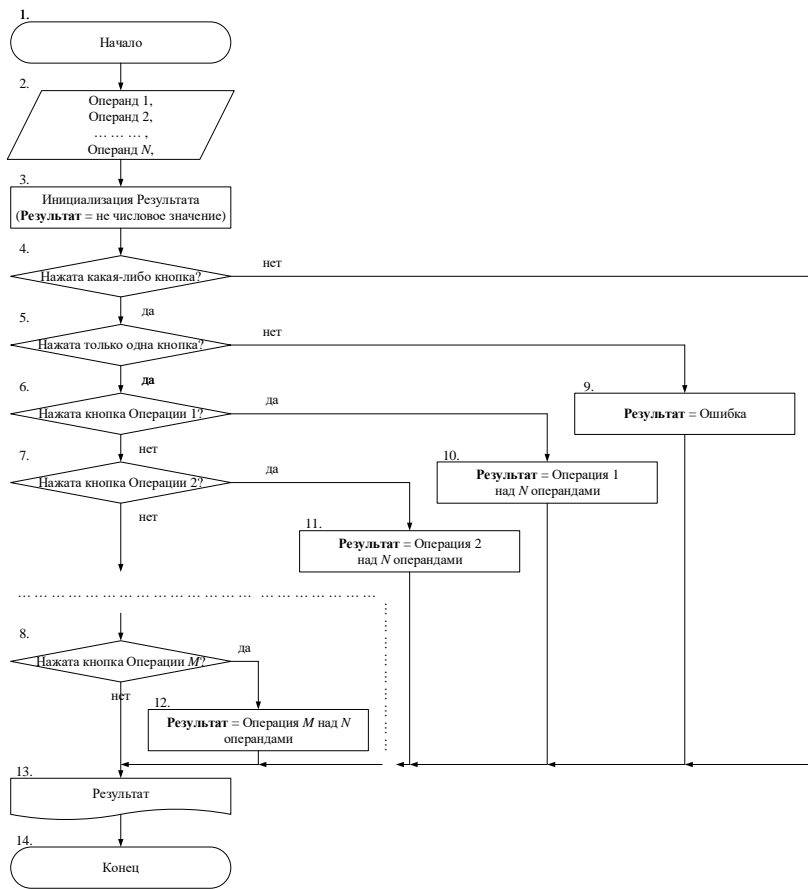


Рисунок 4.7.1 – Схема алгоритма для ВП



#### Описание алгоритма, представленного на Рисунке 4.7.1:

В блоке 1 – обозначено начало вычислительного процесса, обрабатываемого ВП для поиска решения задачи.

В блоке 2 – указан перечень операндов, необходимых для решения поставленной задачи. В общем случае количество операндов – « $N$ ».

В блоке 3 – инициализация – важный шаг для потоковых вычислений, позволяющий оценить, был ли запущен виртуальный прибор или нет? Прежде всего, это важно для ветвей алгоритма, в которых не предусмотрено выполнение никаких действий. Например, ВП открывается с выставленными нулевыми значениями. Запуск на исполнение ВП без инициализации в случае, когда хотя бы одно из условий проведения расчёта не выполняется, приводит к сохранению нулевых показателей и к тому, что пользователю остаётся не ясно: «Запускался ли ВП?», «Содержит ли блок-диаграмма ВП какой-либо исполняемый код?», «Что он сделал не так?», «Нажимал ли он на кнопку запуска?», «Был ли отклик на щелчок "мышью"?» Чтобы снять эти и многие другие вопросы предлагается выполнить инициализацию результата к не числовому значению («NaN»).

В блоке 4 – выполняется проверка нажатия хотя бы одной кнопки операций. Если ни одна из операций (которых в общем случае « $M$ ») не выбрана, то и считать с использованием ВП нечего – переход к блоку 13. Если какая-то из кнопок операций нажата – необходимо понять, какая же именно? – Переход к блоку 5.

В блоке 5 – выполняется проверка того, можно ли однозначно распознать выбранную операцию? Однозначность даёт единственная нажатая кнопка. Потому если сразу нажато несколько кнопок – должна быть сгенерирована ошибка, а именно переход в блок 9. Если однозначность распознавания выбранной операции гарантирована, то выполняется переход к блоку 6.

В блоке 6 – проверяется, выбрана ли операция 1, если это именно она, то выполняется переход к выполнению этой операции над операндами в блок 10. Если это какая-то иная операция – переход к блоку 7.

В блоке 7 – проверяется, выбрана ли операция 2, если это именно она, то выполняется переход к выполнению этой операции над операндами в блок 11. Если это какая-то иная операция – переход к блоку 8.

В блоке 8 – проверяется, выбрана ли операция «М» (последняя), если это именно она, то выполняется переход к выполнению этой операции над операндами в блок 12. Иной исход – это нераспознанная операция и ничего при этом делать не нужно – переход к блоку 13. Отсюда следует, что нет необходимости проверять все «М» операций. Достаточно проверить «М – 1» операцию и если все «М – 1» не удалось распознать, то выполнить над операндами последнюю («М») операцию.

В блоке 9 – в качестве результата записывается сигнал ошибки (константа, которая заранее определена автором разработки как сигнал об ошибке). Как правило, все численные сигналы об ошибках прописываются для пользователя на передней панели ВП в разделе справочной информации.

В блоке 10 – в качестве результата записывается значение, получаемое в ходе выполнения над операндом (операндами) первой операции из списка.

В блоке 11 – в качестве результата записывается значение, получаемое в ходе выполнения над операндом (операндами) второй операции из списка.

В блоке 12 – в качестве результата записывается значение, получаемое в ходе выполнения над операндом (операндами) последней операции («М»).

В блоке 13 – передача значения, сформированного в качестве результата, на соответствующий элемент вывода на передней панели ВП.

В блоке 14 – обозначено завершение вычислительного процесса, обрабатываемого ВП для поиска решения задачи.

## Список литературы:

1. Монахов, О. И. Методика организации Учебной, Производственной, Научно-исследовательской и Преддипломной практик обучающихся : учебно-методическое пособие для бакалавров направления 27.03.04 «Управление в технических системах» / О. И. Монахов, А. И. Сафронов, Л. Н. Логинова, Н. Н. Зольникова, А. Е. Ермакова. – М.: РУТ (МИИТ). – 2020. – 112 с.
2. Сафронов, А. И. Составление отчётной документации по решённым задачам алгоритмизации и программирования: Учебно-методическое пособие для проведения аудиторных занятий по Учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – М.: РУТ (МИИТ). – 2018. – 83 с.
3. Сафронов, А. И. Получение первичных профессиональных умений и навыков научно-исследовательской деятельности: Сборник задач для проведения аудиторных занятий по Учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – М.: РУТ (МИИТ). – 2019. – 91 с.
4. Бузурманкулова, Ч. М. Численное интегрирование дифференциального уравнения перегрева жил токопроводов относительно температуры окружающего мира в среде *LabView* / Ч. М. Бузурманкулова // Проблемы автоматики и управления. – 2021. – № 2 (41). – С. 11-17.
5. Нсуе, Х. М. Б Цифровой умножительно-преобразовательный метод измерения неустойчивости частоты колебаний с применением среды программирования *LabView* / Х. М. Б. Нсуе, В. П. Федосов, С. В. Кучерявенко // Известия ЮФУ. Технические науки. – 2020. – № 3 (213). – С. 55-68.
6. Рияд, А. Программирование *LabView* для анализа вибрации шпинделя станка с ЧПУ / А. Рияд, А. А.

- Молчанов // Инновационные научные исследования. – 2021. – № 2-3 (4). – С. 6-16.
7. Шишленин, А. Е. Автоматизация расчёта установившегося режима электроэнергетической сети на базе *LabView* / А. Е. Шишленин, Е. М. Веселова // В сборнике: Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований. Материалы III Всероссийской национальной научной конференции студентов, аспирантов и молодых ученых. В 3-х частях. Редколлегия: Э.А. Дмитриев (отв. ред.) [и др.]. – 2020. – С. 420-421.
  8. Карамуллин, Т. Х. Разработка информационно-измерительного комплекса системы энергоснабжения космического аппарата на базе *LabView* / Т. Х. Карамуллин, Н. К. Джамалов // Наука и образование сегодня. – 2020. – № 3 (50). – С. 12-14.
  9. Кабаев, Т. К. Разработка виртуального прибора в среде *LabView* для исследования характера переходного процесса в электрических цепях / Т. К. Кабаев, Н. А. Абыкаева // Проблемы автоматики и управления. – 2021. – № 1 (40). – С. 3-11.
  10. Пильцов, М. В. Создание виртуальных лабораторных стендов с помощью *NI LabView* / М. В. Пильцов, В. В. Тихонов // Современные технологии и научно-технический прогресс. – 2021. – № 8. – С. 141-142.
  11. Пильцов, М. В. Обработка видеоизображений многофазных систем в *NI LabView* / М. В. Пильцов, В. В. Тихонов // Современные технологии и научно-технический прогресс. – 2021. – № 8. – С. 141-142.
  12. Википедия. Свободная энциклопедия [Электронный ресурс] : Операнд. URL: <https://ru.wikipedia.org/wiki/Операнд> (дата обращения: 28.07.2021).
  13. Википедия. Свободная энциклопедия [Электронный ресурс] : Операция (программирование). URL:

- [https://ru.wikipedia.org/wiki/Операция\\_\(программирование\)](https://ru.wikipedia.org/wiki/Операция_(программирование)) (дата обращения: 28.07.2021).
14. Центр ПСС – центр проектирования сетей и сооружений [Электронный ресурс] : Перевести Кельвины в Цельсии онлайн калькулятор. URL: <https://www.center-pss.ru/klk/k24.htm> (дата обращения: 24.08.2021).
  15. Каталог онлайн калькуляторов | Онлайн калькулятор [Электронный ресурс] : Градус Цельсия °C в Фаренгейта °F | Онлайн калькулятор. URL: <https://allcalc.ru/converter/celsius-fahrenheit> (дата обращения: 24.08.2021).
  16. Калькуляторы онлайн, справочник-энциклопедия [Электронный ресурс] : Градус Цельсия в градус Ранкина, калькулятор онлайн, конвертер. URL: <https://www.calc.ru/gradus-tselsiya-v-gradus-rankina.html> (дата обращения: 24.08.2021).
  17. Новые калькуляторы – Калькулятор онлайн [Электронный ресурс] : Конвертер из Цельсиев (°C) в Реомюры (°Re) и обратно онлайн. URL: <https://kalku.ru/tselsiy-reaumur-converter/> (дата обращения: 24.08.2021).
  18. Главная страница - *Wikipedia, the free encyclopedia* – Википедия [Электронный ресурс] : Шкала Рёмера – *Römer scale* – Википедия. URL: [https://wikichi.ru/wiki/Römer\\_scale](https://wikichi.ru/wiki/Römer_scale) (дата обращения: 24.08.2021).
  19. *Metric Conversion charts and calculators* [Электронный ресурс] : Конвертация из Градусы Цельсия в Ньютон. URL: <https://www.metric-conversions.org/ru/temperature/celsius-to-newton.htm> (дата обращения: 24.08.2021).
  20. Википедия. Свободная энциклопедия [Электронный ресурс] : Градус Делиля – Википедия. URL: [https://ru.wikipedia.org/wiki/Градус\\_Делиля](https://ru.wikipedia.org/wiki/Градус_Делиля) (дата обращения: 24.08.2021).

21. Википедия. Свободная энциклопедия [Электронный ресурс] : Градуировка – Википедия. URL: <https://ru.wikipedia.org/wiki/Градуировка> (дата обращения: 24.08.2021).
22. Википедия. Свободная энциклопедия [Электронный ресурс] : Постоянная Планка – Википедия. URL: [https://ru.wikipedia.org/wiki/Постоянная\\_Планка](https://ru.wikipedia.org/wiki/Постоянная_Планка) (дата обращения: 24.08.2021).
23. Википедия. Свободная энциклопедия [Электронный ресурс] : Безразмерная величина – Википедия. URL: [https://ru.wikipedia.org/wiki/Безразмерная\\_величина](https://ru.wikipedia.org/wiki/Безразмерная_величина) (дата обращения: 24.08.2021).
24. Сивухин, Д. В. Общий курс физики. В 5-ти томах. Том 4. Оптика / Д. В. Сивухин. – М.: Физматлит. – 2021. – 792 с.
25. Википедия. Свободная энциклопедия [Электронный ресурс] : Гравитационная постоянная – Википедия. URL: [https://ru.wikipedia.org/wiki/Гравитационная\\_постоянная](https://ru.wikipedia.org/wiki/Гравитационная_постоянная) (дата обращения: 24.08.2021).

## **Приложение 1. Типовая структура отчёта по работе, классифицируемой как задача программирования и алгоритмизации (с графическим пользовательским интерфейсом)**

Отчёт по выполненной работе программирования и алгоритмизации должен содержать:

0. Титульный лист.
1. Формулировку цели работы.
2. Описание задачи согласно выданному варианту.
  - 2.1. Общая часть.
  - 2.2. Индивидуальная часть (если есть).
3. Составление блок-схемы алгоритма программы.
  - 3.1. Общая часть.
  - 3.2. Индивидуальная часть (если есть).
4. Подбор и расчёт тестовых примеров.
  - 4.1. Общая часть.
  - 4.2. Индивидуальная часть (если есть).
5. Листинг кода составленного программного обеспечения (иллюстрация блок-диаграммы ВП *National Instruments LabView*).
  - 5.1. Общая часть.
  - 5.2. Индивидуальная часть (если есть).
6. Графический пользовательский интерфейс программного обеспечения (передняя панель ВП *National Instruments LabView*) и его описание.
  - 6.1. Общая часть.
  - 6.2. Индивидуальная часть (если есть).
7. Расчёт тестовых примеров с использованием составленного ВП.
  - 7.1. Общая часть.
  - 7.2. Индивидуальная часть (если есть).
8. Формулировку вывода о проделанной работе (обезличено – исключить из вывода местоимения, такие как «я», «мы» и другие).

## **Приложение 2. Перечень типовых рекомендаций для демонстрации самостоятельности выполненной работы обучающимися**

Сформулирован краткий перечень типовых рекомендаций, уместных для оформления отчёта по каждой из решённых обучающимися задач, изложенных в Сборнике:

1. Блок-схема алгоритма: размер для всех блоков по ширине подбирается исходя из ширины наиболее наполненного текстом блока.

2. Блок-схема алгоритма: для случаев, когда в блоке, согласно пункту 1, необходимо разместить довольно большое количество текста, который в итоге обусловит неоправданно громоздкую схему, рекомендуется часть текста оформить в виде комментария к блоку; размеры блоком-комментариев могут превышать принятые размеры основных блоков (комментарии не нормируются).

3. Блок-схема алгоритма: блоки по высоте не нормируются, однако рекомендуется выполнять увеличение их высоты только пропорционально количеству записанных в них действий. **Иначе**: для всей блок-схемы принимается некоторая средняя высота одного блока, достаточная для описания одного действия, и если блок содержит описание двух действий, то его высота может быть увеличена вдвое, трёх действий – втрое, и так далее.

4. Блок-схема алгоритма: в тех случаях, когда блок-схема, размещаемая на странице А4, масштабируется настолько, что текст в блоках становится нечитаемым, рекомендуется распределить блоки на нескольких страницах А4, используя ссылки для демонстрации переходов между страницами по линиям связи.

5. Листинг кода программы: код программы должен содержать значимые имена переменных или таблицу соответствия, раскрывающую смысловое содержание



введённых автором переменных, не обладающих значащими именами.

6. Листинг кода программы: для удобства навигации по коду автора программы и проверяющего во время защиты работы, код необходимо снабжать комментариями. Комментарии, выполненные в коде, могут также играть роль связки с блок-схемой алгоритма и упрощать проверку отчёта.

7. Подбор тестовых примеров: минимальное количество тестовых примеров, доказывающих работоспособность составленного программного обеспечения, должно быть достаточным для покрытия всех ветвей разветвляющегося вычислительного процесса.

8. Общее: необходимо выполнение дополнительных скриншотов/иллюстраций для случаев, когда текстовое описание проделанных автором работы действий становится громоздким или трудным для подробного текстового описания.

9. Общее: рекомендуется выполнять нумерацию рисунков (если они есть) с подписями, содержащими наименование этих рисунков, например, «Рисунок 1 – Блок-схема алгоритма простого калькулятора, созданного в *National Instruments LabView*».

### **Приложение 3. Перечень требований, предъявляемых к именам отчётных файлов, направляемых на удалённую проверку (по электронной почте)**

**Общий вид формата имени файла:** *«Дата. Задание. Фамилия.docx»*

**Формат записи даты:** *«ГГГГММДД»*, где *ГГГГ* – четыре цифры текущего года, *ММ* – две цифры текущего месяца, *ДД* – две цифры текущего дня.

**Формат записи задания:** *«Задание NNk»*, где *NN* – две цифры номера задания, *k* – обозначение «о», если файл содержит общую часть; обозначение «и», если файл содержит индивидуальную часть; обозначение «ои», если файл содержит как общую, так и индивидуальную части.

**Если устранить замечания по работе удаётся в тот же день:** после фамилии ставится пробел и в круглых скобках записывается номер попытки исправления.

**Примеры правильных имён файлов, которые сдаются на проверку впервые:**

*«20210919. Задание 02о. Иванов.docx»*

*«20210926. Задание 03и. Иванов.xlsx»*

*«20211003. Задание 04ои. Иванов.vi»*

*«20211101. Задание 05ои. Иванов.vsdх»*

**Примеры правильных имён файлов, которые сдаются на проверку повторно в тот же день:**

*«20210919. Задание 02о. Иванов (1).docx»*

*«20210926. Задание 03и. Иванов (1).xlsx»*

*«20211003. Задание 04ои. Иванов (1).vi»*

*«20211101. Задание 05ои. Иванов (1).vsdx»*

## Оглавление

Введение.....	3
1 Начало работы с пакетом прикладных программ <i>National Instruments LabView</i> .....	6
1.1 Переход к созданию нового ВП.....	6
1.2 Описание рабочих окон ВП: передней панели и блок-диаграммы.....	10
1.3 Настройка отображения кириллических шрифтов на блок-диаграмме и передней панели ВП.....	11
1.4 Отображение / сокрытие основных элементов управления.....	13
1.5 Размещение интерфейсных элементов управления на передней панели ВП.....	17
1.6 Размещение функций на блок-диаграмме ВП.....	21
1.7 Создание образов интерфейсных элементов управления на блок-диаграмме ВП.....	28
1.8 Интерпретация связи интерфейсных элементов управления с блок-диаграммой ВП.....	33
1.9 Типы значений и типы связей.....	33
2 Задание «Разработка ВП "Простой калькулятор"».....	45
2.1 Цель работы.....	47
2.2 Полезные соотношения для выполнения задания....	47
2.2.1 Термины.....	47
2.2.2 Связь между численным и логическим элементами управления ВП.....	47
2.3 Пример выполнения задания.....	56
2.3.1 Общая часть.....	57
2.3.2 Индивидуальная часть (переменные).....	58
2.3.3 Индивидуальная часть (константы).....	59
2.4 Варианты индивидуального задания.....	62
2.4.1 Переменные в ВП.....	62
2.4.2 Константы в ВП.....	64
2.5 Пример компоновки графического пользовательского интерфейса ВП.....	71
2.5.1 Общая часть.....	71

2.5.2	Переменные в ВП .....	72
2.5.3	Константы в ВП .....	73
2.6	Рекомендации по размещению информации на блок- диаграмме ВП .....	74
2.6.1	Общая часть.....	74
2.6.2	Переменные в ВП .....	75
2.6.3	Константы в ВП .....	76
2.7	Блок-схемы алгоритмов для типовых решений.....	77
2.7.1	Общая часть.....	77
2.7.2	Переменные в ВП .....	79
2.7.3	Константы в ВП .....	81
3	Задание «Разработка ВП "Конвертор температуры"» .....	83
3.1	Цель работы .....	85
3.2	Полезные соотношения для выполнения задания.....	85
3.2.1	Формулы перевода температурных единиц .....	85
3.2.2	Приведение шкалы к единому графическому уровню.....	86
3.2.3	Инвертирование заполнения шкалы .....	88
3.2.4	Настройка начального значения задающего элемента управления.....	90
3.3	Пример выполнения задания.....	91
3.3.1	Смена режима с элемента вывода на элемент ввода .....	91
3.3.2	Изменение диапазона шкалы виртуального термометра.....	96
3.3.3	Переход к прецизионным измерениям .....	97
3.3.4	Программирование простейшей зависимости в обратной пропорции .....	100
3.3.5	Настройка цветового оформления виртуального термометра.....	101
3.4	Варианты индивидуального задания .....	103
3.5	Пример компоновки графического пользовательского интерфейса ВП .....	108
3.6	Рекомендации по размещению информации на блок- диаграмме ВП .....	110
3.7	Блок-схемы алгоритмов для типовых решений.....	112

4 Задание «Разработка ВП "Умный калькулятор (CASE-калькулятор)"» .....	116
4.1 Цель работы .....	118
4.2 Полезные соотношения для выполнения задания....	118
4.2.1 Основные численные функции, размещаемые на блок-диаграммах ВП.....	118
4.2.2 Использование объединённой арифметики ....	124
4.2.3 Вывод на интерфейсный элемент значений физических постоянных величин, содержащих размерность (единицы измерения).....	126
4.2.4 CASE-структура.....	136
4.3 Пример выполнения задания.....	137
4.3.1 Реализация ВП с выводом различных констант на базе CASE-структуры.....	137
4.3.2 Реализация ВП на базе оператора выбора с двумя операндами и тремя операциями .....	151
4.4 Варианты индивидуального задания .....	156
4.5 Пример компоновки графического пользовательского интерфейса ВП .....	164
4.6 Рекомендации по размещению информации на блок-диаграмме ВП.....	165
4.7 Блок-схемы алгоритма для типового решения .....	167

Список литературы: .....	170
--------------------------	-----

Приложение 1. Типовая структура отчёта по работе, классифицируемой как задача программирования и алгоритмизации (с графическим пользовательским интерфейсом).....	174
Приложение 2. Перечень типовых рекомендаций для демонстрации самостоятельности выполненной работы обучающимися.....	175
Приложение 3. Перечень требований, предъявляемых к именам отчётных файлов, направляемых на удалённую проверку (по электронной почте) .....	177

УЧЕБНО-МЕТОДИЧЕСКОЕ ИЗДАНИЕ

Сафронов Антон Игоревич

Проектирование и создание виртуальных приборов *National  
Instruments LabView*

Сборник типовых задач  
для проведения аудиторных занятий по Учебной практике

Изд. № 58-21